



ESCUELA TÉCNICA SUPERIOR DE INGENIEROS INDUSTRIALES Y DE TELECOMUNICACIÓN

Titulación:

INGENIERO EN INFORMÁTICA

Título del proyecto:

SISTEMA INFORMÁTICO DE GESTIÓN WEB PARA LA
ASOCIACIÓN NAVARRA DE SÍNDROME DE DOWN

Elena Leoz Laquidain

José Javier Astrain Escola

Pamplona, 25 de Noviembre de 2010

Índice

CAPÍTULO 1. INTRODUCCIÓN	3
1.1.- ASOCIACIÓN NAVARRA DE SÍNDROME DE DOWN.....	3
1.1.1.- <i>Objetivos</i>	4
1.1.2.- <i>Actividades</i>	4
1.2.- ESTADO DEL ARTE.....	5
1.3.- OBJETO DEL PROYECTO	6
1.4.- PLANTEAMIENTO DE LA SOLUCIÓN	7
1.5.- FASES EN LA REALIZACIÓN DEL PROYECTO	10
CAPÍTULO 2. ANÁLISIS DE REQUISITOS.....	11
2.1.- REQUISITOS HARDWARE	12
2.2.- REQUISITOS SOFTWARE.....	12
2.2.1.- <i>Requisitos de Apariencia y GUI</i>	12
2.2.2.- <i>Requisitos No Funcionales</i>	13
2.2.3.- <i>Requisitos Funcionales</i>	14
2.3.- DIAGRAMAS DE CASOS DE USO	25
2.4.- DIAGRAMA ENTIDAD-RELACIÓN.....	73
CAPÍTULO 3. DISEÑO.....	90
3.1.- ARQUITECTURA DEL SISTEMA	90
3.2.- DISEÑO RELACIONAL DE LA BASE DE DATOS	93
3.3.- DISEÑO DE LA INTERFAZ.....	120
CAPÍTULO 4. IMPLEMENTACIÓN	128
4.1.- ELEMENTOS NECESARIOS PARA LA IMPLEMENTACIÓN.....	128
4.1.1.- <i>Gestor de contenidos Joomla</i>	129
4.1.2.- <i>Lenguaje de programación PHP</i>	131
4.2.- PROCEDIMIENTO DE PROGRAMACIÓN Y EJEMPLO FUNCIONALIDAD “GESTIÓN DE PERSONAL”	133
4.3.- IMPLEMENTACIÓN “MENÚ”	168
CAPÍTULO 5. CONCLUSIONES Y LÍNEAS FUTURAS	174
BIBLIOGRAFÍA.....	178

Capítulo 1. Introducción

Este primer capítulo pretende dar a conocer la entidad objeto del proyecto que nos ocupa. Es decir, pretende acercarnos a la Asociación Navarra de Síndrome de Down [DownNavarra] realizando una breve presentación de la misma así como de los objetivos que persigue y de las actividades que realiza. Por otro lado, esta introducción va a incidir en el estado actual de la Asociación en cuanto a gestión de información se refiere, lo que llevará consigo la presentación del objetivo que se persigue con la realización del presente proyecto. Finalmente, una vez conocida la situación y el objetivo del proyecto, se realizará un planteamiento de la solución a implementar, así como de las fases necesarias para llevarla a cabo.

A continuación vamos a centrarnos detenidamente en todos y cada uno de los temas abordados en este capítulo, comenzando por la presentación de la entidad Asociación Navarra de Síndrome de Down.

1.1.- Asociación Navarra de Síndrome de Down

La Asociación Navarra de Síndrome de Down es una entidad sin ánimo de lucro que trabaja por las personas con Síndrome de Down mediante acciones que contribuyen a mejorar su calidad de vida y a la defensa de sus derechos, basándose en el principio de igualdad de derechos entre las personas sin discapacidad y las discapacitadas, logrando la inclusión social de estas últimas. Fue constituida en 1991 por iniciativa de un grupo de padres y madres de personas con Síndrome de Down que buscaban soluciones a los problemas particulares que conlleva esta discapacidad. El 20 de Noviembre de 2008 fue declarada de utilidad pública por el Ministerio del Interior y actualmente se encuentra federada en la Federación Española de Instituciones para el Síndrome de Down (FEISD) [DownNavarra] [DownEspaña].

1.1.1.- Objetivos

Una vez presentada de manera general la entidad para la cual se desarrolla el proyecto, en este apartado se van a indicar los principales objetivos que persigue la citada Asociación siempre teniendo como propósito la inclusión social y la igualdad de las personas con Síndrome de Down [DownNavarra].

- Diseñar y participar en las políticas de actuación encaminadas a conseguir la normalización social y mejora de la calidad de vida de las personas con Síndrome de Down.
- Promover intercambios y encuentros entre profesionales, asociaciones y cuantas personas y entidades trabajen sobre temas relacionados con el Síndrome de Down, posibilitando el conocimiento de las diversas experiencias que se estén llevando a cabo y sean beneficiosas para las personas con Síndrome de Down.
- Combatir cualquier forma de discriminación contra las personas con Síndrome de Down, exigir la igualdad de trato en todos los ámbitos de la vida, fomentar su vida independiente y el respeto a su intimidad y dignidad.
- Exigir que la atención a las personas con Síndrome de Down se centre en la persona y no en la discapacidad.
- Establecer programas y acciones encaminadas a promover estudios científicos o sociales sobre el Síndrome de Down y promover proyectos relacionados con la infancia, educación, integración escolar, trabajo, integración laboral, juventud, empleo en general, empleo con apoyo, deporte, ocio, vida adulta e independiente, tutela así como a la atención de gravemente discapacitados.

1.1.2.- Actividades

Para finalizar con la presentación de la Asociación Navarra de Síndrome de Down, se va a realizar un breve repaso por las actividades que realiza, actividades que pueden centralizarse en dos finalidades diferentes [DownNavarra].

Por un lado se encuentran las actividades dirigidas a la sensibilización y sensibilización de la población en general para conocer todos los aspectos relativos a las personas con Síndrome de Down y favorecer su integración social. Entre estas actividades podrían citarse las relacionadas con el contacto con unidades de barrio, con otras asociaciones o centros escolares, siendo esta última de vital importancia ya que es la encargada de establecer un contacto y coordinación continua con los profesionales de los centros donde se encuentran escolarizados alumnos con Síndrome de Down con el objeto de intercambiar experiencias educativas y recursos materiales para alcanzar un mayor desarrollo personal de las personas con Síndrome de Down.

Por otro lado se encontrarían las actividades extraescolares específicas para los chicos y chicas con Síndrome de Down que fomentan la relación entre ellos y entre los padres a través del intercambio de experiencias. En este grupo de actividades podrían citarse las siguientes:

- Grupo de padres. Este programa se basa en el intercambio de experiencias y apoyo entre padres que tienen hijos en edades similares con apoyo de profesionales.
- Informática. El objetivo de esta actividad es formar a personas con Síndrome de Down en el uso del ordenador y de las nuevas tecnologías para facilitar su acceso social, educativo y laboral ya que, a través del aprendizaje de conceptos informáticos, se refuerzan habilidades como la comunicación, habilidades sociales, toma de decisiones o habilidades académicas funcionales.
- Lecto-escritura. A través de esta actividad se pretende proporcionar los medios y las estrategias para alcanzar un desarrollo integral a través de la construcción de significados por medio de la escritura y lectura comprensiva.
- Logopedia. Esta actividad tiene como objetivo el desarrollo de las habilidades lingüísticas, receptivas y expresivas con el fin de estimular el potencial de comunicación de las personas con Síndrome de Down.
- Música. El aprendizaje musical posibilita la comunicación y ayuda a mejorar la expresión en aquellas personas necesitadas de ello, facilita la apertura emocional y el acceso hacia el lenguaje. En esta actividad destaca la creación de un grupo de percusión formado íntegramente por personas con Síndrome de Down.
- Deporte. La práctica deportiva aporta un sinnúmero de beneficios a las personas con discapacidad ya que les ayuda a mejorar y desarrollar su psicomotricidad, fomenta el compañerismo y la memorización de las técnicas específicas de cada especialidad y enseña a trabajar en equipo.
- Ocio y tiempo libre. El ocio es uno de los aspectos cruciales para la integración social de las personas con discapacidad por lo que la Asociación trabaja esta actividad realizando tareas con monitores de tiempo libre.
- Proyecto ITACA. El proyecto “ITACA” es una propuesta de la Asociación Navarra de Síndrome de Down para dar respuesta a las necesidades laborales y educativas de los jóvenes con Síndrome de Down.

1.2.- Estado del arte

Una vez realizada la presentación de la Asociación objeto del proyecto, en este segundo apartado de la introducción se va a exponer la situación actual de la misma en cuanto a sistemas de información se refiere.

Actualmente, tal y como se comprobó en las reuniones mantenidas con el personal de la Asociación al comienzo del proyecto, la Asociación Navarra de Síndrome de Down, dispone de una BBDD Access [MsftAcc] explotada a través de formularios generados con el mismo Access como principal herramienta de gestión de información. Dicha herramienta fue desarrollada al comienzo de la andadura de la entidad ajustándose a las necesidades de la misma en ese momento, pero no fue avanzando al mismo ritmo que lo hizo la Asociación lo que ha provocado que se disponga de una herramienta incompleta que no permite gestionar toda la información manejada en la

Asociación. Por este motivo, el personal de la Asociación se vio obligado a buscar alternativas a través de las cuales poder llevar a cabo la gestión de la información que no se mantenía en la herramienta Access. Así, nos encontramos con que la Asociación realiza gran parte de su trabajo de administración y gestión a través de hojas de cálculo o documentos de texto como complemento a las carencias de los formularios Access.

La situación que acabamos de describir, es decir, el hecho de que la Asociación Navarra de Síndrome de Down mantenga toda su información de manera desestructurada y no se disponga de un único sistema de gestión, provoca que el trabajo administrativo de la Asociación se convierta en una labor tediosa y complicada puesto que, en la mayoría de las ocasiones, requiere que la información se mantenga en tres sistemas diferentes lo que, al mismo tiempo, conlleva el mayor problema derivado de la desestructuración y es que los datos gestionados sean incompletos por posibles olvidos en el mantenimiento de los tres sistemas y, lo que es más problemático, que no sean correctos ni fiables.

Además de los problemas relacionados con la corrección y la fiabilidad de los datos manejados por la Asociación debidos a la carencia de un sistema único y actualizado de gestión de información que ya se han mencionado en el párrafo anterior, debemos indicar en este momento que el mayor problema derivado de la redundancia de datos provocada por el hecho de tener que mantener la información en tres sistemas diferentes es que dicha información sea inconsistente. Es decir, el hecho de que la información gestionada por la Asociación se almacene de forma redundante en tres sistemas diferentes provoca que, en ocasiones, dicha información no se encuentre igual en los tres sistemas ya que ha podido ser actualizada en uno, pero no en todos en los que se encuentra lo que propicia que las consultas sobre la misma información no siempre proporcionen el mismo resultado con el consiguiente equívoco para el personal que realiza la consulta.

Por este motivo, la característica principal del proyecto que nos ocupa es la de organizar la información existente aunándola en un único sistema de gestión que evite la redundancia de información y, por lo tanto, las inconsistencias haciendo que la información sea correcta, consistente y de utilidad para el trabajo que se lleva a cabo en la Asociación. Por otra parte, la realización del proyecto persigue que se permita un mejor aprovechamiento de la información existente haciendo que su explotación sea correcta, más ágil y eficiente, lo que va a asegurar que los resultados obtenidos respondan a la realidad de la Asociación.

1.3.- Objeto del proyecto

Después de haber detallado en el apartado anterior la situación informática en la que se encuentra la Asociación Navarra de Síndrome de Down, en las siguientes líneas se va a exponer una breve reseña sobre el objeto del proyecto que estamos tratando.

El objeto del proyecto es la realización del análisis y diseño de un sistema informático de ayuda para la gestión de información. En concreto, se trata de diseñar una herramienta Web que pueda servir de apoyo en la realización del trabajo llevado a cabo por la Asociación Navarra de Síndrome de Down.

La herramienta diseñada debe ser capaz de integrar la información ya existente para facilitar las labores de gestión administrativa, de recursos y de contenidos llevada a cabo por la Asociación, así como de ofrecer una mejor explotación de la misma. En concreto, la herramienta de gestión que se diseñe debe permitir gestionar el personal que participa en la Asociación (ya sean usuarios, socios, voluntarios o profesionales) así como los programas y actividades que se propongan desde la misma.

Además, deberá permitir gestionar las facturas derivadas de la actividad de la Asociación incorporando un gestor documental que facilite las labores de gestión, así como las subvenciones concedidas a la misma por las diferentes instituciones o entidades colaboradoras. Asimismo, la herramienta será capaz de gestionar los proveedores con los que trabaja la Asociación y los gastos generados por la misma.

Finalmente, la herramienta Web a desarrollar deberá posibilitar la consulta sobre todos los elementos involucrados en la Asociación y cuya gestión también es parte del objeto del proyecto.

Como ya se ha citado al comienzo de este apartado, el objetivo principal del proyecto pasa por realizar el análisis y diseño de una aplicación Web que permita gestionar de manera correcta y eficiente la información existente en la Asociación, por lo que no se considera dentro del alcance de este trabajo la construcción o implementación de dicho sistema aunque sí que se realizará como parte del proyecto la elaboración de un pequeño prototipo que facilite la familiarización con la herramienta que se implementará a futuro y que permita evaluar el alcance y funcionamiento de la misma.

1.4.- Planteamiento de la solución

Llegados a este punto, y después de haber presentado en los dos apartados anteriores la situación informática actual de la Asociación Navarra de Síndrome de Down y el objetivo principal del proyecto que se está tratando, en esta sección se va a realizar una breve presentación de las características generales de la aplicación que se va a diseñar.

- Debe ser una herramienta multiusuario que controle el acceso a través de nombre de usuario y contraseña.
- La herramienta debe disponer de un control de permisos que determine las secciones de la aplicación a las que puede acceder cada usuario.
- A través de la aplicación deben poder gestionarse todas las personas que forman parte de la Asociación. Es decir, la aplicación debe permitir gestionar los alumnos, profesionales, voluntarios y socios existentes en la entidad en cuanto a información administrativa se refiere.
- La herramienta debe permitir gestionar los programas que ofrece la Asociación cada curso académico, así como las actividades impartidas y la relación entre programas y actividades.

- Relacionado con la característica anterior, es fundamental que la aplicación permita gestionar las actividades realizadas en cada momento por cada alumno de la Asociación, así como la participación de profesionales y/o voluntarios en las mismas.
- La aplicación a diseñar debe permitir la gestión de las entidades que colaboran con la Asociación, así como de las subvenciones concedidas por cada una de ellas. Además, es importante que la herramienta permita llevar un control de las facturas y nóminas de profesionales que se presentan a cada subvención cada año.
- De lo expuesto en el punto anterior se deduce que la aplicación debe permitir gestionar tanto las facturas que llegan a la Asociación como los salarios recibidos por los profesionales existentes en la misma.
- También relacionado con los profesionales que imparten actividades en la Asociación, la aplicación debe permitir gestionar las actividades extraordinarias que éstos pueden realizar.
- Debe permitir gestionar y mantener un histórico de los proveedores con los que la Asociación mantiene relaciones contractuales, así como de sus transacciones comerciales.
- Finalmente, la aplicación debe permitir gestionar los gastos extras derivados de las actividades realizadas por profesionales y voluntarios, así como un histórico de la cuota que debe aportar cada año todo socio de la Asociación.

Atendiendo a las características generales que debe satisfacer la aplicación a implementar, la disposición en menús o secciones de la misma podría realizarse de la siguiente manera:

- **Menú Personas:** donde se gestionarán los datos personales de los alumnos, socios, profesionales y voluntarios de la Asociación, así como sus altas y bajas en la misma.
- **Menú Programas:** donde se gestionarán tanto los programas ofrecidos históricamente por la Asociación como los nuevos programas que se ofrecen cada año, siendo posible obtener un listado de las actividades vinculadas a cada programa.
- **Menú Actividades:** donde se gestionarán todas las actividades ofrecidas por la Asociación. En este menú se incluye, además de la gestión de la información genérica de la actividad (horario, lugar, etc.), la vinculación de actividades a programas, la gestión de los grupos de cada actividad con sus fechas de alta y baja y la asociación de profesionales y/o voluntarios a las actividades también con sus periodos de alta y baja.
- **Menú Usuarios-Actividades:** donde se realizarán las matriculaciones de los alumnos en las diferentes actividades existentes, así como las anulaciones de dichas matriculas.

- **Menú Entidades:** donde se gestionará la información relativa a cada entidad que colabora con la Asociación, siendo posible obtener un listado de las subvenciones concedidas a la Asociación por cada una de ellas.
- **Menú Subvenciones:** donde se gestionarán todas las subvenciones concedidas a la Asociación y donde se llevará un control de las facturas y salarios de profesionales presentados a cada subvención.
- **Menú Proveedores:** donde se gestionarán los proveedores que suministran a la Asociación y donde se podrá obtener un listado de las facturas expedidas por los mismos.
- **Menú Facturas:** donde se gestionarán todas las facturas recibidas por la Asociación.
- **Menú Salarios:** donde se gestionarán los sueldos mensuales que recibe cada profesional vinculado con la Asociación.
- **Menú Gastos extraordinarios:** donde se gestionarán los gastos que debe hacer frente la Asociación derivados de las actividades realizadas por profesionales y voluntarios.
- **Menú Actividades extraordinarias:** donde se gestionarán las actividades extraordinarias que puede realizar cada uno de los profesionales existentes en la Asociación.

Además de los menús ya presentados que responden a las necesidades funcionales de la aplicación a implementar, la herramienta contará con las siguientes opciones adicionales a través de las cuales se podrá llevar a cabo la gestión de los usuarios (trabajadores de la Asociación) que podrán acceder a la herramienta, así como de las secciones de la misma a las que podrán acceder.

- **Menú Perfiles:** donde se gestionarán los distintos perfiles de usuario que podrán acceder a la herramienta. Es decir, a través de esta sección se mantendrán los perfiles que indicarán los accesos que tendrán los usuarios (en función del perfil al que pertenezcan) sobre cada una de las opciones del menú existentes en la aplicación.
- **Menú Personal:** donde se gestionará el personal de la Asociación que podrá acceder a la herramienta. De esta manera, a través de esta sección se gestionarán los usuarios con los que el personal accederá a la aplicación y se realizará la vinculación de cada usuario a uno de los perfiles existentes.
- **Menú Cambio Password:** donde se permitirá que cada uno de los usuarios con acceso a la aplicación gestione su propia contraseña de acceso a la herramienta.

Siguiendo la propuesta de solución presentada en este cuarto apartado de la introducción, será posible implementar una herramienta que satisfaga el objeto del proyecto indicado en el apartado anterior ya que se dispondrá de una aplicación que permita realizar una gestión automatizada de la información existente en la Asociación.

1.5.- Fases en la realización del proyecto

Por último, y para finalizar con el capítulo de introducción, vamos a describir las fases que se han llevado a cabo para realizar con éxito el análisis, diseño y construcción de un prototipo de la herramienta de gestión para la Asociación Navarra de Síndrome de Down.

Para comenzar con la realización del proyecto se mantuvieron varias reuniones con el personal administrativo de la Asociación con el objetivo de tomar conocimiento sobre la labor desempeñada por la misma, así como de las tareas administrativas que es necesario llevar a cabo y cómo se realizan hasta el momento. Asimismo, el objetivo de las reuniones incluía la presentación por parte de la Asociación del objetivo que pretendían alcanzar con la realización de este proyecto de manera que se pudiera tener conciencia del trabajo que debiera realizarse.

Después de conocer la situación actual de la Asociación y la situación a la que se pretende llegar una vez finalizado el proyecto, fue necesario realizar un estudio detallado del objetivo perseguido por la Asociación para extraer y acordar de manera conjunta los requisitos y alcance del proyecto.

Una vez definidos y acordados los requisitos funcionales que debe satisfacer la aplicación Web a desarrollar, la tercera fase del proyecto consistió en realizar el diseño del diagrama Entidad-Relación a partir del cual se construiría el modelo relacional de la BBDD de la aplicación. Dentro de esta tercera fase del proyecto se incluyó también la implementación de la BBDD a la que daba lugar el modelo relacional antes elaborado.

Tras definir los requisitos y diseñar y construir la BBDD de la aplicación, se llevó a cabo el análisis orientado a objetos, a través de la realización de los diagramas de casos de uso, en el que quedaron reflejadas todas las opciones funcionales de la herramienta a diseñar.

A continuación, y para comenzar la fase final del proyecto, se realizó el diseño de la interfaz gráfica de la herramienta con la que se podría dar por finalizado el objetivo principal del proyecto ya que éste, como ya se ha citado en varias ocasiones, consiste en llevar a cabo el análisis y diseño de una aplicación Web que permita gestionar de manera correcta y eficiente la información existente en la Asociación. Sin embargo, para que los usuarios finales pudieran visualizar y comprobar que la herramienta diseñada cumplirá con las necesidades y requerimientos detectados en la primera fase del proyecto, se llevó a cabo la implementación de un pequeño prototipo, incluyendo fase de pruebas, que permitiera la familiarización con la herramienta final que será implementada a futuro.

Por último, y para dar por finalizado el proyecto, se ha llevado a cabo la redacción de la presente memoria descriptiva de la herramienta desarrollada en la que se pretende reflejar el trabajo realizado a lo largo de todo el proyecto.

Capítulo 2. Análisis de Requisitos

Una vez vista en el capítulo anterior una introducción sobre la Asociación objeto del proyecto, así como de su situación actual en relación a la gestión informática de información y la situación a la que se desea llegar con la realización del presente trabajo, en este segundo capítulo de la memoria descriptiva del proyecto se va a incidir en los requisitos, tanto hardware como software, que debe satisfacer la herramienta a desarrollar. Así mismo, y derivado de los requisitos que se detecten, en este segundo capítulo se incluirá la tarea de realización del análisis orientado a objetos de la herramienta que permitirá obtener una visión detallada de la funcionalidad de la aplicación a través de los diagramas de casos de uso que se lleven a cabo. El capítulo se cerrará con la presentación del diagrama Entidad-Relación que modela las funcionalidades detectadas en la realización del análisis de requisitos y de los diagramas de clase.

Comenzamos la labor de análisis realizando una pequeña introducción sobre el procedimiento que se ha seguido para definir y validar los requisitos, tanto hardware como software, que se van a presentar en este segundo capítulo. Como ya se ha comentado en el último apartado relativo a las fases en la realización del proyecto del capítulo anterior, para comenzar el proyecto fue necesario mantener reuniones con el personal, tanto administrativo como informático, de la Asociación para tomar conciencia del trabajo que desempeñan, así como de su situación informática actual y de la situación ideal a la que se pretende llegar. De la información obtenida en dichas reuniones, así como del estudio de la Base de Datos ya existente en la Asociación que se realizó paralelamente, se extrajeron los requisitos que se consideraba que debía satisfacer la herramienta a desarrollar. Una vez que los requisitos fueron identificados se presentaron al personal informático y administrativo de la Asociación, quien dio el visto bueno sobre los mismos.

Después de haber descrito brevemente el proceso seguido para acordar los requisitos hardware y software exigibles a la aplicación a desarrollar, a continuación se enumeran detalladamente dichos requisitos.

2.1.- Requisitos Hardware

Los requisitos hardware son las características que debe satisfacer un ordenador para poder soportar y/o ejecutar una aplicación [Def.Req.Hw], es decir, los requisitos hardware vienen impuestos por el entorno en el que va a funcionar una herramienta.

En el caso de la aplicación que nos ocupa, vamos a diferenciar entre los requisitos hardware que debe satisfacer la máquina servidora en la que se encuentre alojada la herramienta y los requisitos hardware que deberá satisfacer la máquina a través de la cual los usuarios hagan uso de la misma.

Para que la aplicación desarrollada pueda ser empleada por los usuarios, basta con que la máquina en la que se encuentre alojada sea un ordenador de características medias con sistema operativo Windows y conexión a Internet.

Por su parte, para que cualquier usuario pueda hacer uso de la aplicación será suficiente con que disponga de un ordenador de características medias con sistema operativo Windows y conexión a Internet. Se indica como sistema operativo a disponer Windows debido a que es el más extendido en el mercado actual y de más fácil manejo siendo, además, el sistema operativo que requiere menos conocimientos informáticos para su manejo, característica importante teniendo en cuenta que los usuarios de la herramienta a implementar responden a un perfil con conocimientos básicos de informática. Asimismo, uno de los motivos importantes a la hora de decidir que el sistema operativo a emplear fuera Windows fue el hecho de que se trata del sistema operativo que se emplea actualmente en la Asociación lo que hace que el personal de la entidad ya se encuentre familiarizado con dicho sistema operativo consiguiendo así evitar problemas de adaptación al nuevo entorno una vez implantada la nueva aplicación.

Como se observa, los requisitos hardware exigidos para poder emplear la aplicación son de fácil cumplimiento, lo que hace que la herramienta desarrollada en este proyecto sea accesible para cualquier usuario puesto que hoy un día la mayoría de las empresas y personas cuentan en su poder con un ordenador de características medias.

2.2.- Requisitos Software

Los requisitos software son las características que una aplicación debe satisfacer para cumplir con los objetivos y expectativas esperados de ella. En el caso de la herramienta que nos ocupa vamos a centrarnos en tres tipos de requisitos software como son los requisitos de interfaz, los requisitos no funcionales y los requisitos funcionales.

2.2.1.- Requisitos de Apariencia y GUI

Los requisitos de interfaz, apariencia o GUI se pueden definir como aquellos requisitos que debe cumplir visualmente la aplicación de manera que sea amigable y funcional para el usuario. Entre este tipo de requisitos podemos indicar los siguientes como requisitos a satisfacer por la aplicación:

- La aplicación debe ser sencilla e intuitiva en su manejo, aspecto que se consigue manteniendo un diseño único y común para todas las secciones o menús de la herramienta de manera que sean muy claras las posibles acciones a llevar a cabo y la manera de hacerlas.
- La herramienta debe contar siempre con un menú visible en la parte izquierda de la misma a través del cual acceder a las diferentes opciones de la aplicación. Además, por medio de dicho menú el usuario podrá conocer en todo momento en qué opción de la aplicación se encuentra.
- Las diferentes ventanas que se muestren cuando el usuario acceda a las opciones del menú deben aparecer siempre en el centro de la pantalla sin ocultar el menú y todas deben seguir el mismo patrón de diseño y organización de componentes.
- Será importante que los botones de acción que se presenten en cada una de las ventanas de la aplicación aparezcan siempre en la misma posición, de manera que el usuario se familiarice con ellos y conozca en todo momento a qué zona de la ventana de aplicación debe dirigirse para realizar una determinada acción.
- En las ventanas que muestren resultados de búsquedas, el usuario debe tener siempre visible el criterio de búsqueda al que corresponde el resultado obtenido.
- En todo momento deberá estar disponible la opción de Cerrar Sesión de manera que los usuarios puedan desconectarse y volver a conectarse de manera sencilla.

2.2.2- Requisitos No Funcionales

Los requisitos no funcionales no se refieren a funciones específicas que deba proporcionar la aplicación desarrollada, sino que definen propiedades y restricciones de los servicios o funciones ofrecidas por la misma como pueden ser requisitos relacionados con la fiabilidad, tiempo de respuesta, capacidad de almacenamiento, etc. Asimismo, los requisitos no funcionales también pueden referirse al proceso de desarrollo atendiendo a aspectos tales como lenguajes de programación o metodología empleada en el desarrollo de la herramienta. En general, este tipo de requisitos son más críticos que los requisitos funcionales puesto que si no son satisfechos, la aplicación desarrollada no resulta de utilidad [Def.Req.NoFunc] [Def.Req.NoFuncII].

La aplicación desarrollada deberá satisfacer los siguientes requisitos no funcionales básicos:

- Deberá visualizarse y funcionar correctamente en cualquier navegador, garantizando una velocidad estable de navegación.
- Deberá cumplir con la Ley Oficial de Protección de Datos garantizando la confidencialidad de la información manejada por la herramienta.
- El acceso de los usuarios a la aplicación desarrollada deberá estar controlado a través de nombre de usuario y contraseña.

- Los tiempos de respuesta ofrecidos por la aplicación en la realización de consultas deberán ser reducidos.
- El sistema operativo empleado en el desarrollo de la herramienta será Windows debido a que es el usado por un mayor número de usuarios gracias a su mayor grado de implantación.
- La aplicación va a ser desarrollada empleando el gestor de contenidos Joomla [Joomla] [JoomlaES] debido a la facilidad de diseño y construcción que posibilita gracias a la gran cantidad de componentes para funcionalidades genéricas que ofrece. Los desarrollos a medida necesarios para cumplir con las expectativas de los usuarios serán realizados empleando el lenguaje de programación PHP [PHP] debido a las facilidades que ofrece en cuestiones de implementación.
- El almacenamiento de los datos manejados por la aplicación se realizará en una base de datos MySQL [MySQL] debido a la facilidad de interacción que ofrece con el gestor de contenidos Joomla.

Derivado de los dos últimos requisitos no funcionales citados, se debe indicar que la máquina servidora en la que se encuentre alojada la aplicación deberá disponer de un servidor Web de aplicaciones como Apache [Apache], un servidor de Base de Datos MySQL y un CMS como Joomla. Asimismo, será necesario que la máquina disponga de un entorno PHP puesto que es el lenguaje de programación empleado para desarrollar la aplicación.

2.2.3- Requisitos Funcionales

Para finalizar con la sección relativa a requisitos software vamos a centrarnos en este momento en los requisitos funcionales, es decir, en las funcionalidades que debe proporcionar la aplicación desarrollada para la Asociación Navarra de Síndrome de Down para poder ofrecer una herramienta capaz de gestionar la información manejada por la entidad.

A continuación se enumeran, agrupados en función de los menús o secciones definidas en el apartado 1.4 del presente documento, los requisitos funcionales que, según las reuniones mantenidas con el personal de la Asociación Navarra de Síndrome de Down, debe cumplir la herramienta a desarrollar.

- **Menú Personas**
 - Toda persona existente en la aplicación será usuario, profesional, socio o voluntario, entendiendo como usuario a las personas con Síndrome de Down que acuden a la Asociación.
 - La aplicación permitirá dar de alta nuevas personas o realizar consultas sobre las personas existentes en el sistema. En ningún momento se permitirá eliminar una persona del sistema, solamente se permitirá dar de baja dicha persona de los diferentes roles en los que estuviera asignada.

- Al dar de alta una persona con cualquiera de los roles existentes en el sistema, la aplicación debe asignarle de manera automática un ID autoincrementable y no modificable.
- El DNI de una persona debe ser único.
- La aplicación sólo permitirá dar de alta personas indicando el rol que desempeñan en la aplicación.
- Al dar de alta una persona será obligatorio asignarla, al menos, a uno de los roles existentes en la aplicación.
- La aplicación debe permitir que una misma persona esté asignada al mismo tiempo a diferentes roles.
- Al asignar una persona a un rol la aplicación de manera automática asigna como fecha de alta de esa persona en ese rol la fecha del sistema, permitiendo modificarla.
- Para poder modificar los datos de las personas o para dar de baja una persona en un determinado rol será necesario realizar previamente una consulta. La aplicación permitirá realizar consultas sobre personas filtrando a través de su DNI, de su nombre y apellidos o del rol que desempeñan en la aplicación (en función del rol seleccionado la aplicación permitirá buscar en función del ID que tenga la persona para ese rol seleccionado).
- Las consultas realizadas sobre las personas existentes en el sistema siempre mostrarán los datos genéricos de las personas y los existentes relativos al tipo de rol de cada persona. En el caso de las personas con rol de usuario existirá una sección que mostrará el socio ordinario vinculado a ese usuario y, en el caso de las personas con rol de socio, existirá una sección que mostrará todos los usuarios asociados al socio en cuestión.
- La aplicación permitirá dar de baja una persona en cada uno de los roles a los que estuviera asignada de manera independiente, asignando como fecha de baja la fecha del sistema y permitiendo modificarla.
- Una misma persona puede darse de alta varias veces para cada uno de los roles existentes en la aplicación habiéndose, para ello, dado de baja previamente.
- Las personas con rol de socio pueden ser de dos tipos: ordinarios o colaboradores.
- Las personas con rol de socio ordinario pueden ser familiares de más de un usuario de la asociación y cada usuario de la asociación puede tener más de un familiar como socio.

- Para cada usuario (persona con Síndrome de Down) es obligatoria la existencia, en cada momento, de un único socio ordinario. Por ello, al dar de alta una persona con rol de usuario la aplicación obligará a introducir el socio ordinario que se hará cargo de sus gastos. Para realizar dicha selección se mostrará un listado con todos los socios ordinarios existentes en la Asociación.
- La aplicación, a través de la sección de modificación de los datos de los usuarios, permitirá modificar el socio ordinario responsable de los gastos de cada usuario, pero no se mantendrá un histórico de los socios ordinarios que han sido responsables de los gastos de un determinado usuario a lo largo del tiempo.
- Al crear una persona con rol de socio ordinario, la aplicación debe cargar de manera automática el campo “Cuota” con el valor existente para las cuotas del año en curso. Además, cada vez que se actualice la tabla maestra de cuotas con un nuevo valor o se cree un nuevo registro para las cuotas, la aplicación deberá comprobar si se trata del año en curso y, si es así, actualizará el campo “Cuota” de todos los socios ordinarios con el nuevo valor introducido. El valor de este campo no será modificable.
- La aplicación permitirá la existencia de tantas personas con rol de socio colaborador como se desee. Este tipo de socio, a diferencia de los socios ordinarios, no se encontrarán vinculados a las personas con Síndrome de Down.
- En el caso de las personas con rol de socio colaborador el campo “Cuota” debe ser un campo opcional y modificable en el que insertar el importe con el que colabora la persona en cuestión. Para este tipo de personas este campo deberá mantenerse actualizado de manera manual.
- En caso de que existiera, el CIF de la persona con rol de socio (ordinario o colaborador) deberá ser único.
- No se permitirá que una misma persona sea socio colaborador y ordinario.
- Las personas con rol de profesional existentes en la aplicación podrán ser de dos tipos: internos o externos.

- **Menú Programas**

- La aplicación debe permitir la gestión (alta, consulta y modificación) de los programas ofrecidos por la asociación. La aplicación no permitirá borrar un programa una vez que haya sido dado de alta.
- Los programas deben crearse por cada curso no pudiendo existir, para el mismo curso, dos programas con el mismo nombre.
- Un programa puede tener varias actividades asociadas al mismo o no tener ninguna.

- Para poder modificar los datos de los programas será necesario realizar previamente una consulta. La aplicación permitirá realizar consultas sobre programas filtrando a través de su nombre y del curso académico asociado al programa.
- Las consultas realizadas sobre los programas existentes en el sistema siempre mostrarán los datos genéricos de dichos programas. Además, al mostrar el resultado de la búsqueda existirá una sección adicional asociada a cada resultado que mostrará todas las actividades existentes para el programa en cuestión.

- **Menú Actividades**

- La aplicación debe permitir la gestión (alta, consulta y modificación) de las actividades ofrecidas por la asociación. La aplicación no permitirá borrar una actividad una vez que haya sido dada de alta.
- El nombre de las actividades debe ser único.
- Toda actividad debe estar asociada de manera obligatoria, al menos, a un programa por lo que, al crear la actividad, se debe indicar a qué programa está vinculada. Una vez que una actividad ha sido asociada a un programa no será posible su desvinculación.
- De cada actividad puede existir cada año más de un grupo, pero puede darse el caso de que en un año una actividad no tenga ningún grupo.
- Si para un curso la actividad tiene grupos, dichos grupos deberán crearse en el momento de crear la actividad.
- Al crear un grupo para un curso académico de una determinada actividad, la aplicación asignará de manera automática la fecha del sistema como fecha de alta de dicho grupo y permitirá que el usuario la modifique para indicar la fecha en la que comienza el grupo de la actividad.
- Todo grupo de cada actividad debe contar con, al menos, un profesional o un voluntario que se haga cargo de dicho grupo, siendo posible que un grupo de una actividad cuente con voluntarios y no con profesionales y viceversa. La asociación de voluntarios y/o profesionales a los grupos de las actividades deberá realizarse en el momento de crear un nuevo grupo.
- Los voluntarios y profesionales pueden hacerse cargo de más de un grupo de una actividad, pudiendo existir voluntarios o profesionales que no participen en ninguna de las actividades.
- Al seleccionar un profesional o un voluntario como responsable de un determinado grupo de una actividad, la aplicación de manera automática asignará la fecha del sistema como fecha de alta de dicho profesional en el correspondiente grupo y permitirá que dicha fecha se modifique manualmente por el usuario.

- Para poder modificar los datos de las actividades (datos genéricos, programas, grupos) será necesario realizar previamente una consulta. La aplicación permitirá realizar consultas sobre actividades filtrando a través de su nombre y del curso académico. Desde el resultado de la búsqueda será posible modificar la información de la actividad, los voluntarios y/o profesionales encargados de la misma, así como crear nuevos grupos o modificar los ya existentes.
- Las consultas realizadas sobre las actividades existentes en el sistema mostrarán, además de los datos genéricos de dichas actividades y los programas a los que están asociadas, los datos relativos a la información de las actividades.
- La modificación de los grupos existentes no debe permitir su eliminación, sino sólo darlo de baja para un determinado curso académico de una actividad. Al hacerlo, la aplicación asignará de manera automática como fecha de baja la fecha del sistema, permitiendo su modificación.
- Un mismo grupo de una misma actividad puede darse de alta varias veces para el mismo curso académico habiéndose, para ello, dado de baja previamente.
- Para asignar nuevos profesionales o voluntarios a un grupo de una determinada actividad o dar de baja uno ya existente será necesario, en primer lugar, realizar una búsqueda sobre el grupo deseado. Una vez localizado el grupo la aplicación permitirá añadirle nuevos profesionales o voluntarios o eliminar los ya asignados. Al eliminar un profesional o voluntario como responsable de un grupo, la aplicación de manera automática indicará como fecha de baja la fecha del sistema permitiendo su modificación.
- La aplicación debe permitir que un profesional o voluntario se pueda dar de baja el mismo día que se dio de alta, siempre que las fechas cumplan el requisito de que la fecha de baja sea posterior a la de alta y la nueva fecha de alta sea posterior a la de baja ya existente.
- Un mismo profesional o voluntario puede darse de alta varias veces para cada uno de los grupos de actividades existentes en la aplicación habiéndose, para ello, dado de baja previamente.
- La aplicación debe permitir la existencia de grupos de actividades para los que no haya usuarios matriculados.

- **Menú Usuarios-Actividades**

- La aplicación debe disponer de una sección en la que matricular o deshacer la matrícula a los usuarios de un determinado grupo de una actividad. En dicha sección se debe permitir realizar búsquedas por actividad, curso académico, grupo y usuario.
- Para matricular a un usuario en un determinado grupo de una actividad, se deberá realizar en primer lugar una consulta sobre los grupos de la actividad en cuestión para el curso académico deseado. Una vez localizado el grupo en cuestión la aplicación mostrará un botón que permitirá matricular usuarios en el grupo seleccionado mostrando, para ello, el listado de usuarios existentes en el sistema. Al matricular un usuario en un grupo, la aplicación mostrará de manera automática como fecha de alta la fecha del sistema y permitirá modificarla.
- Un mismo usuario podrá darse de alta varias veces en un mismo grupo de una actividad habiéndose, para ello, dado de baja previamente.
- La aplicación permitirá dar de baja a un usuario en cada uno de los grupos de actividades en los que estuviera matriculado. Para ello, en primer lugar será necesario realizar una búsqueda por actividad y curso y acceder al grupo en cuestión para visualizar el listado de usuarios matriculados. Una vez seleccionado el usuario deseado se podrá dar de baja asignando como fecha de baja la fecha del sistema y permitiendo modificarla.

- **Menú Entidades**

- La aplicación debe permitir gestionar las entidades con las que trabaja la Asociación. Es decir, debe permitir dar de alta nuevas entidades y modificar los datos de las ya existentes. En ningún caso se debe permitir eliminar una entidad que ya hubiera sido dado de alta.
- El nombre de la entidad debe ser un valor único.
- Para poder modificar los datos de las entidades con las que trabaja la Asociación será necesario realizar previamente una consulta. La aplicación permitirá realizar consultas sobre entidades filtrando a través de su nombre.
- Desde el listado resultante de la búsqueda por entidad, la aplicación debe permitir obtener un listado con todas aquellas subvenciones que hayan sido concedidas por la citada entidad. Dicho listado deberá ser filtrable en función del año.

- **Menú Subvenciones**

- La aplicación debe permitir gestionar las subvenciones solicitadas por la Asociación. Es decir, debe permitir dar de alta nuevas subvenciones y modificar los datos de las ya existentes. En ningún caso se debe permitir eliminar una subvención que ya hubiera sido dado de alta.
- Para poder modificar los datos de las subvenciones gestionadas por la Asociación será necesario realizar previamente una consulta. La aplicación permitirá realizar consultas sobre subvenciones filtrando a través del nombre de la subvención o del año de solicitud y de la entidad que la hubiera concedido.
- Para poder asignar una factura a una determinada subvención, en el detalle de la subvención para el año deseado, se presentará un botón “Asignar factura” a través del que se podrán seleccionar las facturas deseadas. Para cada factura, se deberá indicar el porcentaje de la factura asignado así como el importe correspondiente y la fecha de asignación que, por defecto, será la fecha del sistema.
- Una vez que una factura ha sido asignada a una determinada subvención la aplicación no debe permitir eliminar dicha vinculación. Sólo se debe permitir modificar el porcentaje y el importe asignado a la subvención. Para ello, se deberá realizar previamente una búsqueda por subvención y año. Al realizar la modificación, la fecha debe actualizarse a la fecha del sistema pudiendo ser modificada.
- Toda factura puede presentarse a más de una subvención, pero pueden existir facturas que no se hayan presentado a ninguna subvención. La aplicación deberá comprobar que la suma de los importes concedidos a cada subvención por una determinada factura no supera el importe total de la factura.
- Al mostrar el listado de las facturas presentadas a una determinada subvención en un año en concreto, la aplicación debe mostrar la suma total de los importes de todas las facturas, así como las cantidades presupuestada, solicitada y concedida para dicha subvención en ese año y el importe restante para alcanzar la cantidad concedida.
- Para poder asignar un salario anual a una determinada subvención, en el detalle de la subvención para el año deseado, se presentará un botón “Asignar salario” a través del que se podrán seleccionar los profesionales cuyo salario se desea asignar. Para cada profesional, se deberá indicar el porcentaje de su salario anual asignado así como el importe correspondiente y la fecha de asignación que, por defecto, será la fecha del sistema.

- Una vez que un salario ha sido asignado a una determinada subvención la aplicación no debe permitir eliminar dicha vinculación. Sólo se debe permitir modificar el porcentaje y el importe asignado a la subvención. Para ello, se deberá realizar previamente una búsqueda por subvención y año. Al realizar la modificación, la fecha debe actualizarse a la fecha del sistema pudiendo ser modificada.
- El sueldo anual de todo profesional puede presentarse a más de una subvención, pero pueden existir salarios que no se hayan presentado a ninguna subvención. La aplicación deberá comprobar que la suma de los importes concedidos a cada subvención por un determinado profesional no supera su salario anual.

- **Menú Proveedores**

- La aplicación debe permitir gestionar los proveedores con los que trabaja la Asociación. Es decir, debe permitir dar de alta nuevos proveedores y modificar los datos de los ya existentes. En ningún caso se debe permitir eliminar un proveedor que ya hubiera sido dado de alta.
- El CIF del proveedor debe ser un valor único.
- Para poder modificar los datos de los proveedores con los que trabaja la Asociación será necesario realizar previamente una consulta. La aplicación permitirá realizar consultas sobre proveedores filtrando a través de su nombre o CIF.
- Desde el listado resultante de la búsqueda por proveedor, la aplicación debe permitir obtener un listado con todas aquellas facturas que hayan sido expedidas por el citado proveedor.

- **Menú Facturas**

- La aplicación debe permitir gestionar las facturas recibidas por la Asociación. Es decir, debe permitir dar de alta nuevas facturas y modificar los datos de las ya existentes. En ningún caso se debe permitir eliminar una factura que ya hubiera sido dado de alta.
- Para poder modificar los datos de las facturas será necesario realizar previamente una consulta. La aplicación permitirá realizar consultas sobre facturas filtrando a través de su ID, del proveedor que la emitió, del nombre de la subvención o del año.

- **Menú Salarios**

- La aplicación debe permitir gestionar los salarios recibidos mensualmente por cada uno de los profesionales de la Asociación. Es decir, debe permitir dar de alta nuevos salarios y realizar consultas sobre los salarios existentes filtrando por profesional, mes y año. En este caso, no se debe permitir eliminar un salario ya creado, pero sí será posible realizar modificaciones sobre el mismo siempre y cuando no hay sido presentado a alguna de las subvenciones concedidas a la Asociación.

- **Menú Gastos Extraordinarios**

- La aplicación debe permitir gestionar los gastos extras de voluntarios y profesionales. Para ello se dispondrá de una sección en el menú donde poder dar de alta, consultar (por tipo de persona, nombre de persona y concepto) y modificar los gastos de dichas personas.

- **Menú Actividades Extraordinarias**

- La aplicación debe permitir gestionar las actividades extras realizadas por los profesionales. Para ello se dispondrá de una sección en el menú donde poder dar de alta, consultar (por tipo de actividad, curso y profesional) y modificar dichas actividades extraordinarias.
- La aplicación debe permitir que un mismo profesional realice actividades extraordinarias diferentes en el mismo curso, pero no que realice la misma actividad varias veces en el mismo curso.

- **Menú Perfiles**

- La aplicación debe permitir gestionar los diferentes tipos de acceso sobre cada una de las secciones del menú a través de la gestión de perfiles de usuario. Con este fin, la aplicación permitirá la creación de nuevos perfiles de usuario, la modificación de los ya existentes y el borrado de un perfil siempre y cuando no exista ningún usuario vinculado al mismo.
- Al crear un nuevo perfil de usuario, el nombre dado al mismo debe ser único.
- Para poder realizar la modificación o el borrado de un perfil existente será necesario realizar previamente una consulta filtrando a través del nombre dado al perfil de usuario.
- Un mismo perfil de usuario puede tener asociados a más de un trabajador de la Asociación, pero puede ser posible que existan perfiles de usuario a los que no se haya vinculado ningún trabajador.

- Para cada una de las secciones del menú de la aplicación las posibilidades de acceso que se ofrecerán serán las siguientes: “Sin acceso”, “Lectura” o “Escritura”. Un usuario cuyo perfil para una determinada sección sea “Sin acceso” visualizará la opción del menú, pero no podrá acceder a la misma, mientras que si su perfil es “Lectura” podrá acceder a dicha sección, pero en modo consulta, es decir, no podrá realizar ninguna modificación sobre la información existente. Finalmente, si el perfil del usuario es “Escritura” podrá realizar todo tipo de acciones y modificaciones sobre la información de la sección para la que se le ha asignado permiso de escritura.
- Cada uno de los perfiles existentes sólo puede tener un tipo de permiso para cada una de las secciones del menú de la aplicación. Es decir, para cada sección, el perfil indicará si no se tiene acceso, si se tiene acceso de lectura o de escritura, pero no será posible que en un mismo perfil para una misma sección se tenga dos tipos de acceso diferentes.
- Al crear un nuevo perfil de usuario la aplicación seleccionará por defecto que dicho perfil tenga permiso de escritura sobre el menú de cambio de password y que no tenga acceso al resto de las secciones del menú.
- La aplicación contará con un perfil “Superadministrador” que será el único que tendrá permiso de escritura sobre todas las secciones del menú.
- La aplicación no permitirá que un perfil diferente al de “Superadministrador” tenga acceso sobre los menús “Perfiles” y “Personal”. Es decir, la única persona que podrá gestionar tanto los perfiles como los usuarios para el personal de la Asociación será el superadministrador. Para el resto de secciones del menú se permitirá que todos los perfiles existentes puedan tener cualquier tipo de acceso.

- **Menú Personal**

- La aplicación debe permitir la gestión (alta, consulta, modificación y borrado) de los distintos usuarios (personal de la Asociación) que podrán acceder a la herramienta.
- Para poder modificar o eliminar alguno de los usuarios con acceso a la aplicación será necesario realizar previamente una consulta. La herramienta permitirá realizar consultas filtrando a través del nombre de usuario, del *login* de acceso a la aplicación o del perfil asignado al usuario en cuestión.
- Todo usuario de la aplicación, entendiendo por usuario al personal de la Asociación, debe estar vinculado de manera obligatoria a uno y solo a uno de los perfiles existentes por lo que, al crear un nuevo usuario se deberá indicar el perfil al que pertenece, siendo posible que posteriormente el usuario se asocie a otro de los perfiles existentes.

- El *login* que da acceso a la aplicación a cada uno de los usuarios debe ser único, es decir, no podrá haber dos usuarios que tengan el mismo *login*.
- La aplicación no permitirá que exista más de un usuario vinculado al perfil de “Superadministrador”. Para el resto de perfiles existentes se permitirá vincular a ellos tantos usuarios como se desee.
- El usuario con perfil de “Superadministrador” no podrá ser eliminado en ningún momento y, como ya se ha comentado anteriormente, será el único que pueda gestionar tanto los perfiles como los usuarios de acceso a la herramienta.

- **Menú Cambio Password**

- La aplicación permitirá que todo usuario (trabajador de la Asociación con acceso a la herramienta) con permisos para ello pueda modificar la contraseña con la que acceder al sistema.

Además de los requisitos agrupados por secciones de la aplicación que ya se han presentado, la herramienta permitirá la gestión de las direcciones físicas en las que residen las personas que forman parte de la Asociación o las entidades que colaboran con la misma, por lo que dicha gestión se llevará a cabo desde las opciones de creación y modificación tanto de personas como de entidades. La gestión de direcciones se llevará a cabo de tal manera que en ningún momento se eliminará una dirección ya creada, sino que solamente será posible la creación de nuevas direcciones y la modificación de las ya existentes. Para ello, la introducción o modificación de los datos relativos a la dirección, tanto desde la creación o modificación de personas como de entidades, deberá realizarse de la siguiente manera: la aplicación permitirá que el usuario introduzca el valor que se desee en cada uno de los campos de la dirección y, conforme se vaya introduciendo texto, mostrará las coincidencias del valor introducido con los datos de las direcciones ya existentes. Así, si el usuario comprueba que el dato que desea introducir ya existe deberá seleccionarlo de la lista de valores que le presente la aplicación, mientras que si no se encuentra disponible deberá continuar con la introducción del mismo manualmente. Finalmente, en el momento de grabar la información de la persona o entidad, la aplicación deberá comprobar si la agrupación de todos los datos vinculados con la dirección existe o no en el sistema y, en función de dicha comparación, crear o no una nueva dirección.

2.3.- Diagramas de Casos de Uso

Después de haber definido detalladamente los requisitos funcionales que debe satisfacer la herramienta a implementar, como tarea a realizar dentro del análisis de requisitos, se va a llevar a cabo el análisis orientado a objetos de la misma que permitirá obtener una visión más detallada de la funcionalidad descrita en los requisitos a través de la elaboración de los diagramas de casos de uso correspondientes. De esta manera, este tercer apartado del capítulo dedicado al análisis de requisitos comenzará con una pequeña introducción sobre los diagramas de casos de uso centrándose a continuación en la presentación de todos y cada uno de los diagramas que describen la funcionalidad de la aplicación desarrollada para la Asociación Navarra de Síndrome de Down.

Los diagramas de casos de uso forman parte de la fase de análisis de una aplicación puesto que documentan el comportamiento del sistema desde el punto de vista del usuario. Es decir, determinan los requisitos funcionales del sistema representando las funciones que puede ejecutar y su interacción con el usuario [Def.Diag.CasoUso]. Como elementos básicos de un caso de uso encontramos los siguientes: [Def.Diag.CasoUso] [Def.Diag.CasoUsoII] [Def.Diag.CasoUsoIII]

- **Actores**

Los actores representan un tipo de usuario del sistema entendiendo por usuario cualquier cosa externa que interactúa con el sistema. Es decir, los actores representan quién o qué inicia una acción dentro del sistema. Por lo tanto, un actor es simplemente un rol que es llevado a cabo por una persona o cosa. Los actores en los diagramas de casos de uso se representan por una figura en forma de persona.

- **Caso de uso**

Un caso de uso es una descripción de la secuencia de interacciones que se producen entre un actor y el sistema cuando el actor emplea el sistema para llevar a cabo una tarea específica. Es decir, un caso de uso representa una funcionalidad concreta dentro del sistema y se representa en el diagrama de casos de uso mediante una elipse con el nombre del caso de uso en su interior.

- **Asociaciones**

Las asociaciones representan las interacciones entre los actores y los casos de uso que reflejan las acciones que los usuarios llevan a cabo a través del sistema.

Para la correcta realización de los diagramas de casos de uso es necesario que cada uno de ellos se acompañe de una descripción textual del mismo en la que quede reflejada la forma de interactuar entre el sistema y el usuario. En dicha descripción debe incluirse la siguiente información: [Desc.Diag.CasoUso]

- **Actores:** Usuarios que interactúan con el sistema.
- **Descripción:** Descripción de la funcionalidad del sistema representada en el caso de uso.
- **Precondiciones:** Hechos que deben cumplirse para que el flujo de evento del caso de uso pueda llevarse a cabo.

- **Flujo de eventos normal:** Descripción de la ejecución normal y exitosa del caso de uso.
- **Flujo de eventos alternativos:** Descripción de los eventos que permiten indicar la forma de actuar del sistema en los casos menos frecuentes e inesperados.
- **Postcondiciones:** Hechos que deben cumplirse si el flujo de eventos normal se ha ejecutado correctamente.

Una vez realizada la introducción a los diagramas de casos de uso se va a llevar a cabo la presentación de aquellos que se han definido para poder describir la funcionalidad del sistema.

2.3.1.- Caso de Uso: Acceso a la herramienta (Iteración 0)

La Figura 1 que se muestra a continuación representa el diagrama de caso de uso correspondiente al acceso del usuario a la aplicación Web desarrollada para la Asociación Navarra de Síndrome de Down.

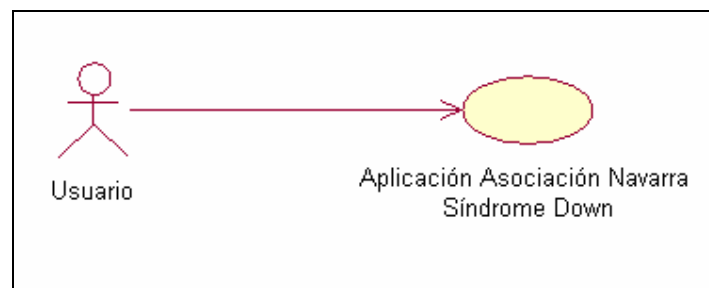


Figura 1. Diagrama de caso de uso: Draft

- **Actores**
El actor del caso de uso es el usuario (personal de la Asociación) que accede a la aplicación.
- **Descripción**
El caso de uso muestra la interacción que debe llevarse a cabo entre el usuario y la aplicación Web para que el usuario pueda acceder a la herramienta a través de su nombre de usuario y contraseña.
- **Precondiciones**
 - El usuario no debe encontrarse ya *logado* en la aplicación.
 - La aplicación debe estar cargada en un navegador Web.
- **Postcondiciones**
 - El usuario podrá utilizar la herramienta correctamente y acceder a los menús a los que tenga acceso.

- **Flujo normal**
 - El usuario introduce la URL de acceso a la aplicación en un navegador Web.
 - El usuario introduce su nombre de usuario y contraseña.
 - El usuario accede a la aplicación Web.
- **Flujos alternativos**
 - El servidor Web no se encuentra disponible y no es posible cargar la aplicación Web en el navegador.
 - El nombre de usuario no se ha introducido.
 - La contraseña no se ha introducido.
 - El nombre de usuario introducido es incorrecto.
 - La contraseña introducida es incorrecta.

2.3.1.1.- Caso de Uso: Menú aplicación Web (Iteración 1)

Una vez que el usuario ha accedido correctamente a la aplicación Web, ésta presentará todas las opciones disponibles para el usuario *logado* a través de un menú. El diagrama de caso de uso para la selección de una de las opciones del menú se muestra en la Figura 2.

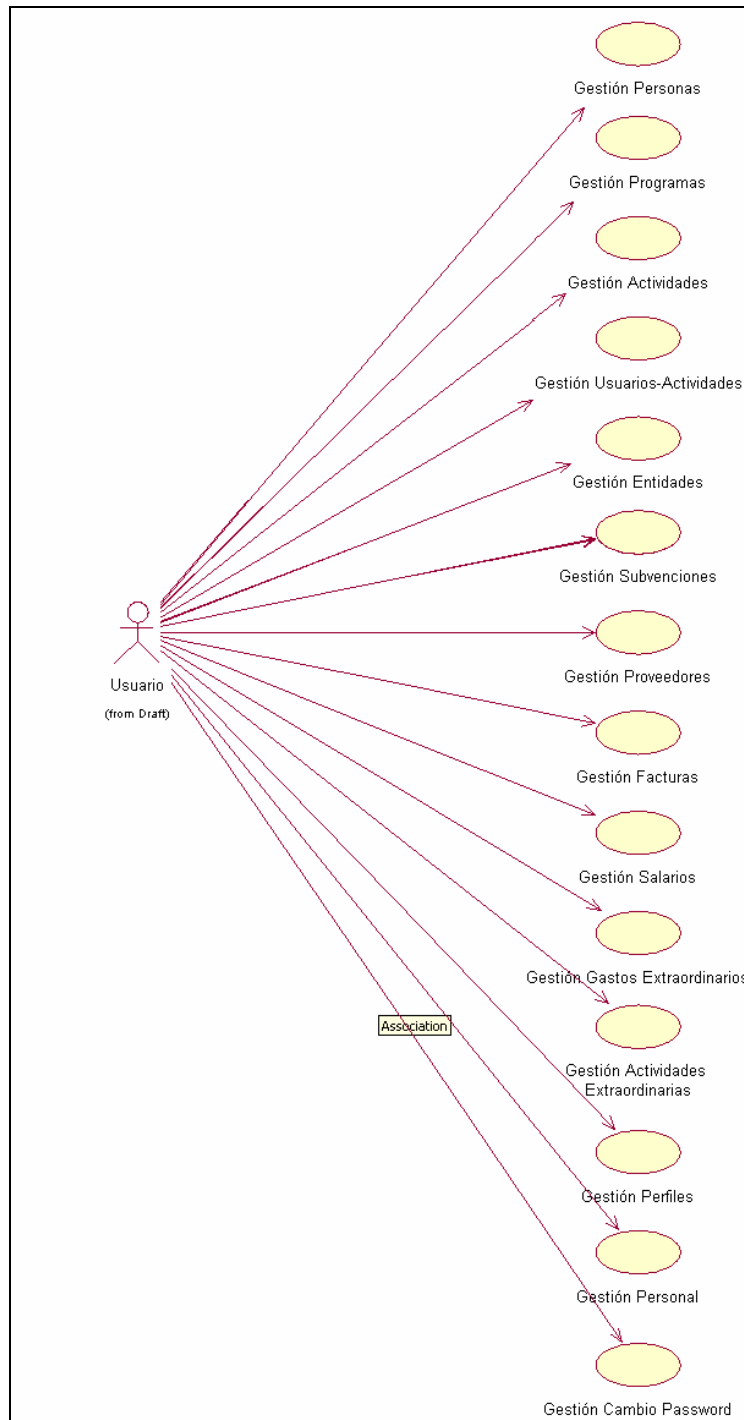


Figura 2. Diagrama de caso de uso Aplicación Asociación Navarra Síndrome Down

- **Actores**
El actor del caso de uso es el usuario (personal de la Asociación) que se encuentra *logado* en la aplicación Web.
- **Descripción**
El caso de uso muestra la situación en la que el usuario ha accedido correctamente a la aplicación Web y debe seleccionar una de las opciones que se le presentan a través del menú de la herramienta.
- **Precondiciones**
 - El usuario se encuentra correctamente *logado* en la aplicación.
- **Postcondiciones**
 - El usuario visualiza el menú de la aplicación con todas las opciones existentes en el mismo. Además, se encontrará disponible la posibilidad de cerrar la sesión iniciada y abandonar la aplicación.
- **Flujo normal**
 - El usuario accede correctamente a la aplicación.
 - La aplicación presenta el menú con todas las opciones existentes en la misma. Si el usuario *logado* en la aplicación tiene perfil “Superadministrador” tendrá acceso a todos los apartados de la herramienta. Sin embargo, si se trata de un usuario normal, tendrá acceso únicamente a los menús para los que se le hayan dado permisos.
 - La aplicación presenta la posibilidad de cerrar sesión para abandonar la herramienta.
- **Flujos alternativos**
 - El servidor Web no se encuentra disponible y no es posible acceder a la aplicación.
 - El usuario intenta acceder a una opción del menú para la que el perfil al que pertenece no tiene acceso.

2.3.1.1.1.- Caso de Uso: Gestión Personas (Iteración 2)

La primera de las opciones del menú disponible para el usuario es la de gestión de personas, donde podrá crear nuevas personas y realizar consultas sobre las ya existentes. Además, a través de la consulta de personas se podrán realizar modificaciones de la información asociada a las mismas, así como dar de alta o de baja personas ya existentes. La Figura 3 que se muestra a continuación representa el diagrama de casos de uso para la gestión básica de personas.

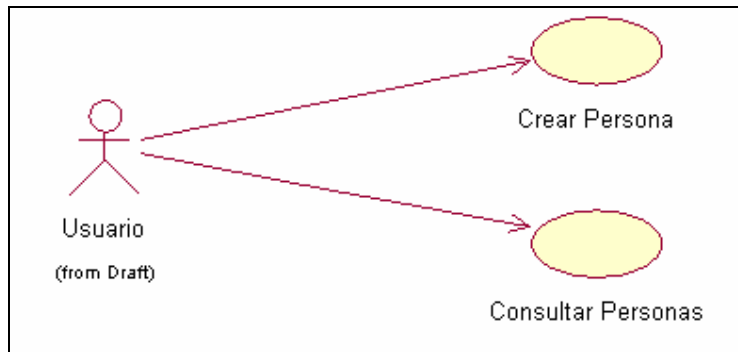


Figura 3. Diagrama de caso de uso: Gestión Personas

- **Actores**
El actor del caso de uso es el usuario (personal de la Asociación) que se encuentra *logado* en la aplicación Web.
- **Descripción**
El presente caso de uso muestra las posibles interacciones entre el usuario y el sistema para llevar a cabo la gestión de las personas. Esto es, muestra las interacciones que deberán realizarse para poder crear nuevas personas o realizar consultas sobre las mismas.
- **Precondiciones**
 - El usuario ha seleccionado la opción *Personas* del menú de la aplicación.
- **Postcondiciones**
 - El usuario visualiza en la parte central de la pantalla un listado de todas las personas existentes en la aplicación.
 - El usuario visualiza en la parte superior de la pantalla los campos por los que es posible realizar una búsqueda de personas (DNI, nombre, apellidos, rol e ID de la persona en el rol correspondiente), además de la opción de crear nuevas personas.

- **Flujo normal**

- El usuario accede a la opción *Personas* del menú de la aplicación.
- El usuario visualiza, en la parte central de la pantalla, el listado actual de personas existentes en la aplicación y, en la parte superior de la misma pantalla, la sección para poder realizar búsquedas sobre las personas ya existentes en el sistema. Además, el usuario dispondrá de los botones para poder crear personas y consultar las ya existentes.
- El usuario selecciona la opción de crear una nueva persona a través del botón disponible para ello en la parte superior de la pantalla.
 - La aplicación mostrará una nueva pantalla donde introducir la información genérica relativa a todas las personas (nombre, apellidos, DNI, dirección, etc.), así como una sección de roles donde poder asignar los roles de la persona en creación.
- El usuario selecciona la opción de realizar una búsqueda sobre las personas existentes en la aplicación a través del botón disponible para ello en la parte superior de la pantalla, habiendo introducido previamente, al menos, un criterio de búsqueda.
 - La aplicación mostrará en la parte central de la misma pantalla un listado mostrando la información de carácter general de todas aquellas personas que satisfagan el criterio de búsqueda introducido.
 - Pulsando sobre la información de una persona en concreto, la aplicación mostrará una nueva pantalla donde se visualizará, además de la información genérica de la persona, la del rol o roles desempeñados por dicha persona.
 - Si el rol ejercido por la persona seleccionada es “Usuario” la aplicación mostrará además una sección donde se visualizará el socio ordinario vinculado a la misma. Si el rol ejercido por la persona seleccionada es “Socio” la aplicación mostrará además una sección donde se visualizarán todos los usuarios con los que se encuentra relacionada.

- **Flujos alternativos**

- El servidor Web no se encuentra disponible y no es posible acceder a la aplicación.
- El usuario intenta acceder a la opción *Personas* del menú para la que el perfil al que pertenece no tiene acceso.
- El usuario intenta crear una nueva persona cuando el perfil de usuario al que pertenece no tiene permisos de escritura sobre la opción *Personas* del menú de la aplicación.
- El usuario no introduce correctamente todos los datos necesarios para poder crear una nueva persona.

- El usuario, en la creación de una nueva persona, no realiza la asociación de la misma con, al menos, uno de los roles existentes.
- El usuario pulsa el botón para realizar una búsqueda sin haber introducido previamente, al menos, un criterio de búsqueda.
- El usuario introduce un criterio de búsqueda para el que no existen personas en la aplicación.

2.3.1.1.1.- Caso de Uso: Crear Persona (Iteración 3)

Dentro de la gestión de personas, como ya se ha visto en el caso de uso anterior, la primera de las opciones disponibles es la relativa a la creación de nuevas personas en la Asociación. A través de dicha creación, como se puede observar en la siguiente figura correspondiente al caso de uso, además de poder introducir los datos genéricos de toda persona, el usuario podrá realizar la asignación de roles a la persona creada así como parte de la gestión de las direcciones físicas manejadas por la aplicación.

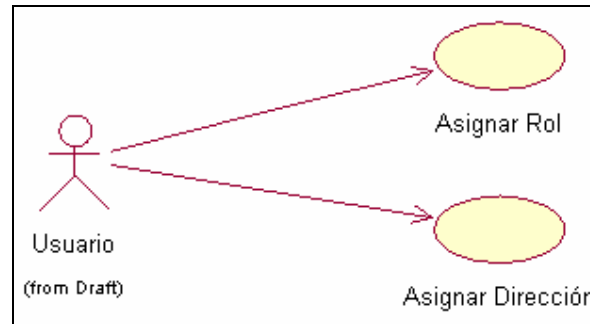


Figura 4. Diagrama de caso de uso: Crear Persona

- **Actores**
El actor del caso de uso es el usuario (personal de la Asociación) que se encuentra *logado* en la aplicación Web.
- **Descripción**
El presente caso de uso muestra las posibles interacciones entre el usuario y el sistema para realizar el alta de una nueva persona que permita que dicha persona entre a formar parte de la herramienta ya sea como profesional, voluntario, socio o usuario (persona con Síndrome de Down) de la Asociación.
- **Precondiciones**
 - El usuario ha seleccionado la opción de crear una nueva persona existente dentro de la sección *Personas* del menú de la aplicación.
- **Postcondiciones**
 - El usuario visualiza en la parte central de la pantalla un listado con todos aquellos campos de información general que es necesario introducir para poder dar de alta una nueva persona. Además, visualizará una sección a través de la cual se podrá realizar la asignación de la persona creada a los diferentes roles existentes en el sistema.

- **Flujo normal**

- La aplicación mostrará una nueva pantalla donde introducir la información genérica relativa a todas las personas (nombre, apellidos, DNI, dirección, etc.) así como una sección de roles donde poder asignar los roles de la persona en creación.

- Pulsando el botón para asignar un rol a la nueva persona la aplicación mostrará una nueva pantalla donde poder seleccionar el rol deseado de entre los disponibles. Al seleccionarlo la aplicación mostrará los datos que es necesario introducir en función del rol seleccionado, además de mostrar como fecha de alta modificable la fecha actual.

- Si el rol seleccionado es “Usuario” la aplicación mostrará, además, un nuevo campo con todos los socios ordinarios existentes en la aplicación para que se seleccione uno de ellos como persona responsable de los gastos del usuario creado.

- Si el rol seleccionado es “Socio” la aplicación mostrará un nuevo campo donde poder indicar si se trata de un socio ordinario o colaborador. En el primer caso la aplicación cumplimentará el campo “Cuota” de manera automática con el valor de cuota existente para el año actual y dicho valor no será modificable mientras que si el socio es de tipo colaborador el campo “Cuota” será libre a rellenar con el dato que se desee.

- Al crear una nueva persona, la aplicación mostrará los datos relativos a la dirección de dicha persona ordenados de manera que el usuario realice la introducción de los mismos del más genérico al más particular (provincia, población, código postal, calle, número, escalera, piso y letra).

- El usuario podrá introducir información en cualquiera de los campos relativos a la dirección y la aplicación, conforme el usuario vaya introduciendo datos, irá mostrando las coincidencias existentes en el sistema con el dato introducido. Así, el usuario podrá seleccionar uno de los valores mostrados o continuar con la introducción manual del dato deseado.

Al grabar los datos relativos a la nueva persona creada la aplicación comprobará si ya existe en el sistema una dirección coincidente con todos los datos introducidos por el usuario. Si es así, no creará una nueva dirección y asignará a la nueva persona creada la dirección ya existente. Por el contrario, si todos los datos introducidos para la dirección de la nueva persona no coinciden totalmente con una ya existente, la aplicación creará una nueva dirección y se la asignará a la persona en cuestión.

- **Flujos alternativos**

- El servidor Web no se encuentra disponible y no es posible acceder a la aplicación.
- El usuario no introduce correctamente todos los datos necesarios para poder crear una nueva persona.
- El usuario, en la creación de una nueva persona, no realiza la asociación de la misma con, al menos, uno de los roles existentes.
- El usuario intenta asignar un rol de socio colaborador a una persona ya relacionada con el rol de socio ordinario o viceversa.

2.3.1.1.2.- Caso de Uso: Consultar Personas (Iteración 3)

La segunda de las opciones existentes dentro de la gestión de personas, como ya se ha visto en el caso de uso correspondiente a dicha gestión, es la de realizar consultas sobre las personas ya existentes en la Asociación. A través de dicha consulta de personas la aplicación permite que los usuarios modifiquen la información de la persona consultada, que la den de baja en un determinado rol o que la den de alta en un nuevo rol o en uno en el que había sido dada de baja previamente. El caso de uso correspondiente a las mencionadas acciones se presenta en la Figura 5.

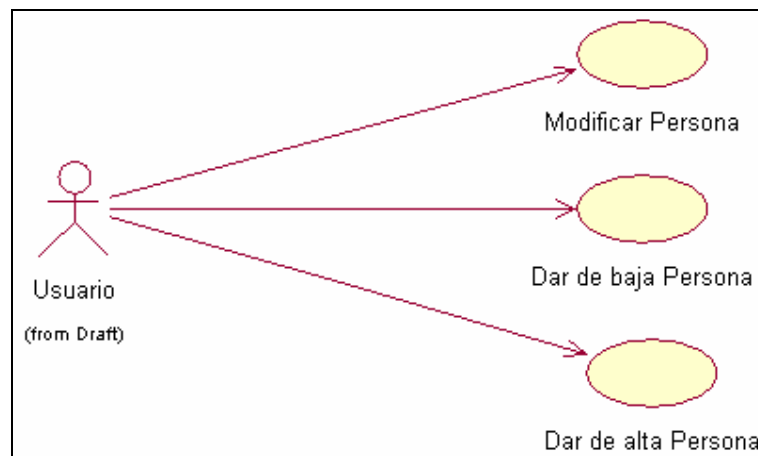


Figura 5. Diagrama de caso de uso: Consultar Personas

- **Actores**

El actor del caso de uso es el usuario (personal de la Asociación) que se encuentra *logado* en la aplicación Web.

- **Descripción**

El presente caso de uso muestra las posibles interacciones entre el usuario y el sistema para llevar a cabo la modificación de una persona o su alta o baja en un determinado rol.

- **Precondiciones**

- El usuario ha realizado de manera correcta una búsqueda de personas a través del buscador existente dentro de la opción *Personas* del menú de la aplicación.

- **Postcondiciones**

- El usuario visualiza en la parte central de la pantalla un listado mostrando la información de carácter general de todas aquellas personas que satisfacen el criterio de búsqueda introducido para realizar la consulta.

- **Flujo normal**

- Una vez localizada la persona deseada en el listado resultante de la búsqueda, el usuario podrá pulsar sobre la información de dicha persona para acceder a una nueva pantalla donde se visualizará, además de la información genérica de la persona, la del rol o roles desempeñados por dicha persona y donde se podrán realizar las modificaciones que se consideren oportunas.
 - Si la persona seleccionada se encuentra dada de baja en alguno de los roles a los que había sido asignada, la aplicación mostrará un botón para poder darla de alta nuevamente. Al pulsar dicho botón la aplicación mostrará una nueva pantalla donde el usuario podrá modificar la información relacionada con el rol así como la fecha de alta en dicho rol que, por defecto, será la fecha actual. Además, el usuario dispondrá de un botón donde poder asignar a la persona seleccionada a un rol al que no hubiera estado asignada previamente.
 - Si la persona seleccionada se encuentra dada de alta en un determinado rol, la aplicación mostrará un botón para poder darla de baja. Al pulsar dicho botón la aplicación mostrará una nueva pantalla donde el usuario podrá modificar la fecha de baja en dicho rol que, por defecto, será la fecha actual.
 - Si la persona seleccionada se encuentra asociada al rol de “Usuario” la aplicación permitirá que se modifique el socio ordinario encargado de los gastos del usuario en cuestión.
 - Si la persona seleccionada se encuentra asociada al rol de “Socio ordinario” la aplicación mostrará una sección con todos aquellos usuarios relacionados con el socio seleccionado. Junto a cada registro se mostrará un botón de borrado a través del que se podrá eliminar la relación entre el socio seleccionado y el usuario asociado al registro. Además, la aplicación mostrará un botón de añadir a través del cual será posible asociar un nuevo usuario a un socio ordinario.

- Si el usuario realiza modificaciones sobre alguno de los campos relativos a la dirección de la persona que se está actualizando, al grabar dichos cambios la aplicación deberá comprobar si la dirección modificada se encuentra asociada, además de a la persona actual, a más personas y/o entidades. Si es ese el caso, deberá mostrar un mensaje informando de tal situación al usuario para que éste determine si desea modificar sólo su dirección o también la del resto de personas y/o entidades vinculadas a la misma dirección. En función de la respuesta dada por el usuario la aplicación modificará la dirección ya existente o no realizará ninguna modificación sobre ella y creará una nueva dirección con la información actual introducida por el usuario.
- **Flujos alternativos**
 - El servidor Web no se encuentra disponible y no es posible acceder a la aplicación.
 - El usuario intenta dar de baja a una persona asignada a un rol “Socio” cuando dicha persona se encuentra asociada a otra persona con rol de “Usuario”.
 - El usuario, al dar nuevamente de alta una persona que se encontraba dada de baja, introduce como fecha de alta una fecha anterior a la fecha en la que la persona se había dado de baja.
 - El usuario, al dar de baja una persona que se encontraba dada de alta, introduce como fecha de baja una fecha anterior a la fecha en la que la persona se había dado de alta.

2.3.1.1.2.- Caso de Uso: Gestión Programas (Iteración 2)

Como ya se ha indicado en varias ocasiones, otra de las opciones disponibles a través de la herramienta es la gestión de los programas ofertados por la Asociación, cuyo diagrama de caso de uso se muestra en la Figura 6.

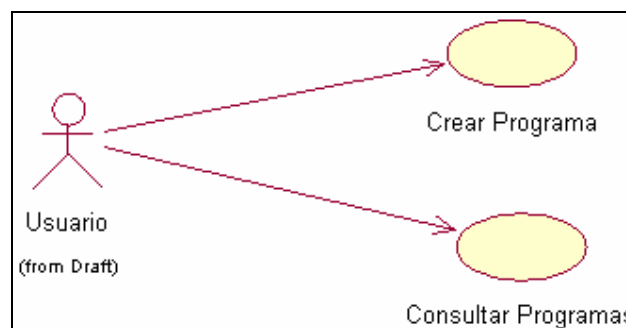


Figura 6. Diagrama de caso de uso: Gestión Programas

- **Actores**

El actor del caso de uso es el usuario (personal de la Asociación) que se encuentra *logado* en la aplicación Web.

- **Descripción**

A través del presente diagrama de caso de uso se pretende dar una visión de las interacciones que el usuario deberá realizar con el sistema tanto para poder crear nuevos programas como para realizar consultas sobre los ya existentes.

- **Precondiciones**

- El usuario ha seleccionado la opción *Programas* del menú de la aplicación.

- **Postcondiciones**

- El usuario visualiza en la parte central de la pantalla un listado de todos los programas existentes en la aplicación.
- El usuario visualiza en la parte superior de la pantalla los campos por los que es posible realizar una búsqueda de programas (nombre del programa o curso en el que se realizó), además de la opción de crear nuevos programas.

- **Flujo normal**

- El usuario accede a la opción *Programas* del menú de la aplicación.
- El usuario visualiza en la parte central de la pantalla el listado actual de programas existentes en la aplicación, así como las opciones de creación y búsqueda.
- El usuario selecciona la opción de crear un nuevo programa a través del botón disponible para ello en la parte superior de la pantalla.
 - La aplicación mostrará una nueva pantalla donde introducir la información del programa a crear (nombre, curso, responsable y descripción) así como una sección de actividades donde poder asignar al programa en creación todas las actividades que lo constituyan.
- El usuario selecciona la opción de realizar una búsqueda sobre los programas existentes en la aplicación a través del botón disponible para ello en la parte superior de la pantalla, habiendo introducido previamente, al menos, un criterio de búsqueda.
 - La aplicación mostrará en la parte central de la misma pantalla un listado mostrando la información de carácter general de todos los programas que satisfagan el criterio de búsqueda introducido.

- **Flujos alternativos**

- El servidor Web no se encuentra disponible y no es posible acceder a la aplicación.
- El usuario intenta acceder a la opción *Programas* del menú para la que el perfil al que pertenece no tiene acceso.
- El usuario intenta crear un nuevo programa cuando el perfil de usuario al que pertenece no tiene permisos de escritura sobre la opción *Programas* del menú de la aplicación.
- El usuario no introduce correctamente todos los datos necesarios para poder crear un nuevo programa.
- El usuario pulsa el botón para realizar una búsqueda sin haber introducido previamente, al menos, un criterio de búsqueda.
- El usuario introduce un criterio de búsqueda para el que no existen programas en la aplicación.
- El usuario intenta crear, para el mismo curso académico, dos programas con el mismo nombre.

2.3.1.1.2.1.- Caso de Uso: Crear Programa (Iteración 3)

De manera similar al caso de uso correspondiente a la gestión de personas, en el caso de la gestión de programas la creación de un nuevo programa incluye, además de la inserción de información de carácter general sobre el programa que ya se ha visto en el caso de uso anterior (nombre, curso, responsable y descripción) la asignación de actividades al programa en creación, tal y como se observa en el siguiente caso de uso.



Figura 7. Diagrama de caso de uso: Crear Programa

- **Actores**

El actor del caso de uso es el usuario (personal de la Asociación) que se encuentra *logado* en la aplicación Web.

- **Descripción**

El caso de uso actual pretende describir las interacciones que el usuario deberá realizar con el sistema para poder llevar a cabo la asignación de actividades al programa en creación.

- **Precondiciones**

- El usuario ha seleccionado la opción de crear un nuevo programa existente dentro de la sección *Programas* del menú de la aplicación.

- **Postcondiciones**
 - El usuario visualiza en la parte central de la pantalla un listado con todos aquellos campos de información general que es necesario introducir para poder dar de alta un nuevo programa. Además, visualizará una sección a través de la cual se podrá realizar la asignación de las actividades existentes en el sistema al nuevo programa en creación.
- **Flujo normal**
 - La aplicación mostrará una nueva pantalla donde introducir la información del programa a crear (nombre, curso, responsable y descripción) así como una sección de actividades donde poder asignar al programa en creación todas las actividades que lo constituyan.
 - Pulsando el botón para asignar una actividad al nuevo programa, la aplicación mostrará una nueva pantalla donde poder seleccionar las actividades deseadas.
- **Flujos alternativos**
 - El servidor Web no se encuentra disponible y no es posible acceder a la aplicación.
 - El usuario no introduce correctamente todos los datos necesarios para poder crear un nuevo programa.

2.3.1.1.2.2.- Caso de Uso: Consultar Programas (Iteración 3)

Para finalizar con la descripción del caso de uso correspondiente a la gestión de programas, se va a incidir en este momento en el detalle del caso relativo a la consulta de programas ya existentes a través del cual, como se observa en la Figura 8, se podrá llevar a cabo la modificación de la información de un determinado programa o la consulta para obtener el listado de todas aquellas actividades que se encuentran asociadas al programa en cuestión.

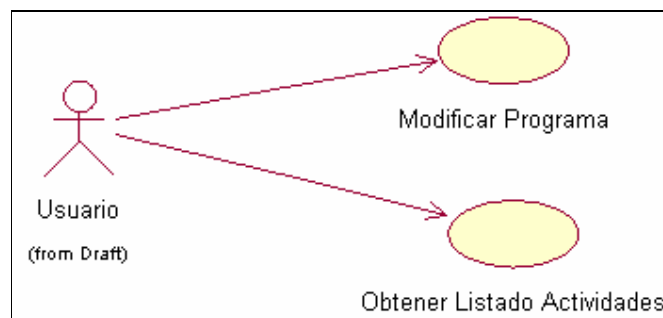


Figura 8. Diagrama de caso de uso: Consultar Programas

- **Actores**
El actor del caso de uso es el usuario (personal de la Asociación) que se encuentra *logado* en la aplicación Web.
- **Descripción**
El caso de uso actual pretende describir las interacciones que el usuario deberá realizar con el sistema para poder realizar una consulta sobre un determinado programa o para obtener el listado de actividades asociadas al mismo.
- **Precondiciones**
 - El usuario ha realizado de manera correcta una búsqueda de programas a través del buscador existente dentro de la opción *Programas* del menú de la aplicación.
- **Postcondiciones**
 - El usuario visualiza en la parte central de la pantalla un listado mostrando la información de carácter general de todos aquellos programas que satisfacen el criterio de búsqueda introducido para realizar la consulta.
- **Flujo normal**
 - Una vez localizado el programa deseado en el listado resultante de la búsqueda, el usuario podrá pulsar sobre la información de dicho programa para acceder a una nueva pantalla donde se visualizará, además de la información genérica del programa, el listado de actividades vinculadas al mismo y desde donde se podrán realizar las modificaciones necesarias relativas tanto a la información general como a la asignación de nuevas actividades al programa.
- **Flujos alternativos**
 - El servidor Web no se encuentra disponible y no es posible acceder a la aplicación.
 - El usuario no introduce correctamente todos los datos necesarios para poder crear un nuevo programa.

2.3.1.1.3.- Caso de Uso: Gestión Actividades (Iteración 2)

La tercera opción del menú existente en la herramienta se corresponde con la gestión de las actividades propuestas desde la Asociación Navarra de Síndrome de Down. La Figura 9 que se presenta a continuación refleja el caso de uso correspondiente a dicha gestión de actividades que, como se puede observar, es idéntico al presentado en la Figura 6 relativa a la gestión de programas ofertados por la Asociación. Por este motivo, la descripción del caso de uso actual se va a realizar de manera genérica haciendo referencia al caso de uso de la gestión de programas.

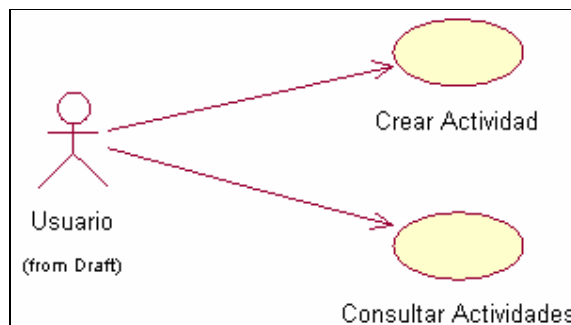


Figura 9. Diagrama de caso de uso: Gestión Actividades

Al igual que en el caso de la gestión de programas, el actor involucrado en la gestión de actividades (creación de nuevas actividades y consulta de las ya existentes) es el usuario (personal de la Asociación) que se encuentre *logado* en la aplicación Web.

De la misma manera, la precondition y postcondiciones asociadas al caso de uso de la gestión de actividades son idénticas a las presentadas para el caso de la gestión de programas con la salvedad de que, en este caso, el usuario accede a la opción *Actividades* del menú y visualiza en la pantalla central el listado de actividades existentes y la posibilidad de crear nuevas actividades o realizar búsquedas sobre las ya creadas filtrando por nombre de actividad o por curso en el que se impartió.

El flujo normal asociado a la gestión de actividades responde a las mismas acciones presentadas en el flujo normal de la gestión de programas con la diferencia de que, en este caso, el usuario accede a la opción *Actividades* del menú de la aplicación y ésta muestra en la parte central el listado de actividades ya existentes junto con las opciones de creación y búsqueda. Al igual que en el caso de la creación de nuevos programas, si el usuario selecciona la opción de crear una nueva actividad se le mostrará una nueva pantalla donde introducir el nombre de la actividad, el programa al que pertenece y los comentarios que se consideren oportunos, además de una sección donde poder asignar a la actividad en creación todos los grupos que la constituyan. De manera similar, el proceso a seguir en el caso de que el usuario seleccione la opción de realizar una búsqueda de actividades es idéntico al presentado para la misma acción en el caso de uso de la gestión de programas.

Finalmente, los flujos alternativos existentes para el caso de uso de la gestión de actividades son los mismos que se presentaron para el caso de uso de la gestión de programas salvando las distancias de que en el aquel caso eran relativos a la sección de *Programas* y ahora lo son a la sección de *Actividades*. Además de estos flujos alternativos idénticos a los de la gestión de programas, la gestión de actividades puede presentar las siguientes situaciones erróneas adicionales:

- El usuario intenta crear una actividad asignándole un nombre ya existente para alguna de las actividades creadas con anterioridad.
- El usuario, al crear la nueva actividad, no indica el programa al que se desea asociar la actividad en creación.

2.3.1.1.3.1.- Caso de Uso: Crear Actividad (Iteración 3)

La Figura 10 que se presenta a continuación hace referencia al caso de uso relativo a la creación de una nueva actividad en el sistema. De igual forma que en el caso de uso anterior, la situación que se presenta en este momento es idéntica a la presentada en el apartado 2.3.1.1.2.1 para la creación de un nuevo programa. Por este motivo se remite a dicha sección para obtener la explicación del caso de uso, siempre teniendo en cuenta que donde se menciona programas en este momento serían actividades y que donde se refiere a la asignación de actividades a los programas en este momento sería la asignación de grupos a la actividad en creación introduciendo toda la información relativa al grupo (lugar, horario, precio, responsables, profesionales y voluntarios). Además, y a diferencia del caso de creación de un nuevo programa, al crear un nuevo grupo para una actividad la aplicación introduce, por defecto, la fecha actual como fecha de alta del grupo y permite que dicha fecha sea modificada manualmente por el usuario *logado* en la herramienta.



Figura 10. Diagrama de caso de uso: Crear Actividad

2.3.1.1.3.1.1.- Caso de Uso: Asignar grupos (Iteración 4)

Dentro del proceso de creación de una nueva actividad, como se ha podido observar en el caso de uso anterior, es posible realizar la asignación de diferentes grupos a la actividad en cuestión. Dentro del proceso de asignación de grupos a actividades se incluye la introducción de información genérica relativa al grupo (lugar, horario, precio, responsables y fecha alta) así como la asignación de profesiones y/o voluntarios como responsables de un determinado grupo. La gestión de esta última asignación de profesionales y/o voluntarios es la que nos ocupa en el presente caso de uso, tal y como se puede observar en la Figura 11 que se muestra a continuación.

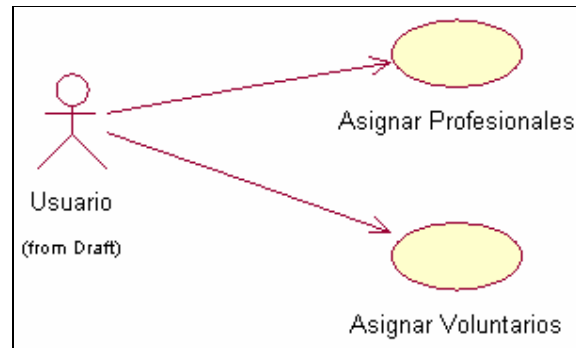


Figura 11. Diagrama de caso de uso: Asignar grupos

- **Actores**
El actor del caso de uso es el usuario (personal de la Asociación) que se encuentra *logado* en la aplicación Web.
- **Descripción**
El caso de uso actual pretende describir las interacciones que el usuario deberá realizar con el sistema para poder llevar a cabo la asignación de profesionales y/o voluntarios a un determinado grupo de una actividad.
- **Precondiciones**
 - El usuario ha seleccionado la opción, accesible desde la creación de una nueva actividad, de asignar un grupo a una determinada actividad.
- **Postcondiciones**
 - El usuario visualiza en la parte central de la pantalla un listado con todos aquellos campos de información general que es necesario introducir para poder crear un nuevo grupo para una determinada actividad (lugar, horario, precio y responsable). Además, visualizará una sección a través de la cual se podrá realizar la asignación de profesionales al grupo en creación y otra sección idéntica para realizar la asignación de voluntarios al mismo grupo.

- **Flujo normal**

- Pulsando el botón para asignar un nuevo grupo a la actividad en creación, la aplicación mostrará una nueva pantalla donde introducir la información del grupo a crear (lugar, horario, precio y responsable) así como una sección de profesionales donde poder asignar al grupo en creación todos los profesionales encargados del mismo y una sección de voluntarios donde poder realizar la misma asignación para los voluntarios encargados del grupo.
 - Pulsando el botón para asignar un profesional al nuevo grupo la aplicación mostrará una nueva pantalla donde poder seleccionar el profesional deseado de entre todos los disponibles. Al seleccionar uno de los profesionales, la aplicación mostrará la fecha actual como fecha de alta de ese profesional como responsable del grupo y permitirá que el usuario modifique dicha fecha.
 - Pulsando el botón para asignar un voluntario al nuevo grupo la aplicación mostrará una nueva pantalla donde poder seleccionar el voluntario deseado de entre todos los disponibles. Al seleccionar uno de los voluntarios, la aplicación mostrará la fecha actual como fecha de alta de ese voluntario como responsable del grupo y permitirá que el usuario modifique dicha fecha.

- **Flujos alternativos**

- El servidor Web no se encuentra disponible y no es posible acceder a la aplicación.
- El usuario no introduce correctamente todos los datos necesarios para poder crear un nuevo grupo.
- El usuario intenta grabar los datos del nuevo grupo sin haber introducido un profesional o voluntario responsable del mismo.

2.3.1.1.3.2.- Caso de Uso: Consultar Actividades (Iteración 3)

Del mismo modo que en el caso de la gestión de personas y programas, la gestión de actividades dispone de la opción para poder realizar consultas sobre las actividades ya existentes en la Asociación. A través de dicha consulta será posible que el usuario modifique la información de la actividad consultada así como de los grupos asociados a la misma, siendo posible dar de baja o de alta un determinado grupo o un determinado profesional o voluntario dentro de un grupo. El caso de uso correspondiente a las mencionadas acciones se presenta en la Figura 12.

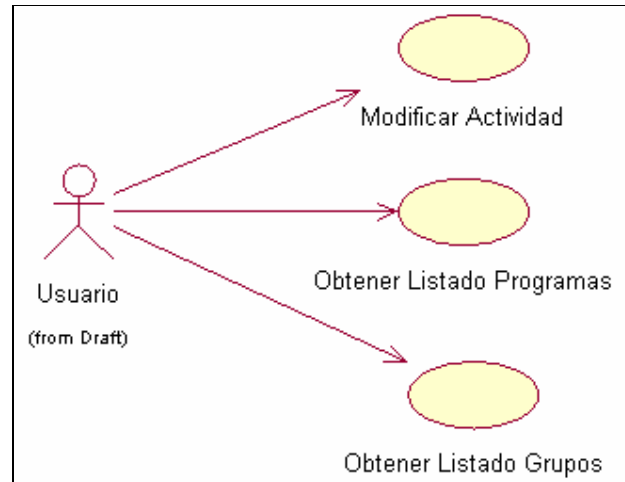


Figura 12. Diagrama de caso de uso: Consultar Actividades

- **Actores**
El actor del caso de uso es el usuario (personal de la Asociación) que se encuentra *logado* en la aplicación Web.
- **Descripción**
El presente caso de uso muestra las posibles interacciones entre el usuario y el sistema para llevar a cabo la modificación de una actividad, de sus grupos o de los profesionales y/o voluntarios asociados a dichos grupos. Además, se incluyen en este caso de uso las interacciones necesarias para poder dar de alta o baja los grupos asociados a las actividades y los profesionales y/o voluntarios vinculados a los citados grupos.
- **Precondiciones**
 - El usuario ha realizado de manera correcta una búsqueda de actividades a través del buscador existente dentro de la opción *Actividades* del menú de la aplicación.
- **Postcondiciones**
 - El usuario visualiza en la parte central de la pantalla un listado mostrando la información de carácter general de todas aquellas actividades que satisfacen el criterio de búsqueda introducido para realizar la consulta.

- **Flujo normal**

- Una vez localizada la actividad deseada en el listado resultante de la búsqueda, el usuario podrá pulsar sobre la información de dicha actividad para acceder a una nueva pantalla donde se visualizará, además de la información genérica de la actividad, la del programa o programas a los que se encuentra asociada dicha actividad así como el listado de grupos existentes para la misma. A través de esta nueva pantalla se podrán realizar sobre las actividades las modificaciones que el usuario considere oportunas (en ningún momento la aplicación permitirá que se modifique el programa o programas a los que se encuentra vinculada una actividad).
- Si la búsqueda realizada sobre las actividades se ha llevado a cabo filtrando únicamente por nombre de actividad, al acceder a una de las actividades existentes en el resultado, la aplicación mostrará el listado de todos los programas y grupos en los que la actividad se encuentre dada de alta, independientemente del año en el nos encontremos. Por el contrario, si la búsqueda se realiza filtrando únicamente por el curso, al acceder a una de las actividades existentes en el resultado, la aplicación mostrará todos los programas y grupos asociados a la actividad para el año sobre el que se realizó la búsqueda. Este último criterio es el mismo que se seguirá en el caso de que la búsqueda se realice filtrando por nombre de actividad y curso académico.
- Para cada uno de los grupos existentes para la actividad que se encuentre dado de baja, la aplicación mostrará un botón para poder darlo de alta nuevamente. Al pulsar dicho botón la aplicación mostrará una nueva pantalla donde el usuario podrá modificar la información relacionada con el grupo así como la fecha de alta de dicho grupo que, por defecto, será la fecha actual.
- Para cada uno de los grupos existentes para la actividad que se encuentre dado de alta, la aplicación mostrará un botón para poder darlo de baja. Al pulsar dicho botón la aplicación mostrará una nueva pantalla donde el usuario podrá modificar la fecha de baja de dicho grupo que, por defecto, será la fecha actual.
- Si el usuario selecciona la opción de crear un nuevo grupo para la actividad, la aplicación mostrará una nueva pantalla donde introducir la información genérica de dicho grupo (lugar, horario, precio y responsable) y donde poder asignar profesionales y/o voluntarios a dicho grupo.
- Al acceder al detalle de cada uno de los grupos de la actividad seleccionada, la aplicación mostrará, además de la información genérica del grupo, dos listados: uno con todos aquellos profesionales vinculados al grupo de la actividad y otro con todos los voluntarios asociados al mismo grupo de la actividad.

- **Flujos alternativos**

- El servidor Web no se encuentra disponible y no es posible acceder a la aplicación.
- El usuario intenta dar de baja todos los grupos existentes para una determinada actividad.
- El usuario no introduce correctamente todos los datos necesarios para poder crear un nuevo grupo para una determinada actividad.
- El usuario, al dar nuevamente de alta un grupo que se encontraba dado de baja, introduce como fecha de alta una fecha anterior a la fecha en la que el grupo se había dado de baja.
- El usuario, al dar de baja un grupo que se encontraba dado de alta, introduce como fecha de baja una fecha anterior a la fecha en la que el grupo se había dado de alta.

2.3.1.1.3.2.1.- Caso de Uso: Obtener listado grupos (Iteración 4)

Para finalizar con los casos de uso existentes para poder llevar a cabo la gestión de las actividades ofertadas por la Asociación Navarra de Síndrome de Down, en este apartado se va a presentar el caso de uso correspondiente a la obtención del listado tanto de profesionales como de voluntarios asociados a un determinado grupo de una actividad. La Figura 13 que se presenta a continuación hace referencia a la obtención de dichos listados a través de los cuales la aplicación va a permitir al usuario realizar las altas y bajas sobre dichos profesionales que se consideren oportunas.

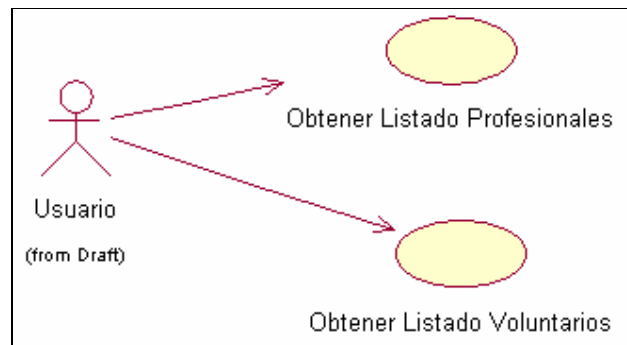


Figura 13. Diagrama de caso de uso: Obtener listado grupos

- **Actores**

El actor del caso de uso es el usuario (personal de la Asociación) que se encuentra *logado* en la aplicación Web.

- **Descripción**

El presente caso de uso muestra las posibles interacciones entre el usuario y el sistema para llevar a cabo las altas y bajas de los profesionales y/o voluntarios asociados a los grupos de las actividades, así como para poder asignar nuevos profesionales y/o voluntarios a los citados grupos.

- **Precondiciones**

- El usuario ha accedido al detalle de uno de los grupos existentes para la actividad seleccionada después de haber realizado de manera correcta una búsqueda de actividades a través del buscador existente dentro de la opción *Actividades* del menú de la aplicación.

- **Postcondiciones**

- El usuario visualiza en la parte central de la pantalla un listado mostrando la información de carácter general del grupo seleccionado, así como un listado de todos los profesionales y voluntarios asociados al citado grupo.

- **Flujo normal**

- Al acceder al detalle de cada uno de los grupos de la actividad seleccionada, la aplicación mostrará, además de la información genérica del grupo, dos listados: uno con todos aquellos profesionales vinculados al grupo de la actividad y otro con todos los voluntarios asociados al mismo grupo de la actividad.
 - Si el grupo para el que se ha accedido al detalle se encuentra dado de baja, la aplicación no permitirá realizar ninguna modificación sobre la información mostrada por pantalla.
 - Si el grupo para el que se ha accedido al detalle se encuentra dado de alta, la aplicación permitirá modificar su información.
- Para cada uno de los profesionales dados de baja en el grupo, la aplicación mostrará un botón para poder darlo de alta nuevamente. Al pulsar dicho botón la aplicación mostrará una nueva pantalla donde el usuario podrá modificar la fecha de alta de dicho profesional que, por defecto, será la fecha actual. El mismo procedimiento se seguirá para el caso de dar de alta un voluntario que se encontrara dado de baja en el grupo.
- Para cada uno de los profesionales dados de alta en el grupo, la aplicación mostrará un botón para poder darlo de baja. Al pulsar dicho botón la aplicación mostrará una nueva pantalla donde el usuario podrá modificar la fecha de baja de dicho profesional que, por defecto, será la fecha actual. El mismo procedimiento se seguirá para el caso de dar de baja un voluntario que se encontrara dado de alta en el grupo.
- Si el usuario selecciona la opción de asignar un nuevo profesional y/o voluntario para el grupo de la actividad, la aplicación mostrará una nueva pantalla donde seleccionar un profesional o voluntario de entre los existentes y donde poder modificar la fecha de alta de dicho profesional o voluntario que, por defecto, será la fecha actual.

- **Flujos alternativos**

- El servidor Web no se encuentra disponible y no es posible acceder a la aplicación.
- El usuario intenta dar de baja todos los profesionales y voluntarios existentes para un determinado grupo de la actividad.
- El usuario, al dar nuevamente de alta un profesional o voluntario que se encontraba dado de baja, introduce como fecha de alta una fecha anterior a la fecha en la que el profesional o voluntario se había dado de baja.
- El usuario, al dar de baja un profesional o voluntario que se encontraba dado de alta, introduce como fecha de baja una fecha anterior a la fecha en la que el profesional o voluntario se había dado de alta.

2.3.1.1.4.- Caso de Uso: Gestión Usuarios-Actividades (Iteración 2)

Relacionada con la gestión de actividades presentada en el apartado anterior, la herramienta diseñada cuenta con una nueva opción a través de la cual será posible realizar la matriculación de un usuario en un determinado grupo de una actividad, así como deshacer la matrícula realizada para un usuario en un grupo de una actividad. Asimismo, a través de esta nueva opción del menú de la aplicación, será posible obtener un listado de todos los usuarios que se encuentran matriculados en cada uno de los grupos existentes para cada actividad. El caso de uso correspondiente a la gestión de usuarios matriculados en actividades que se acaba de describir es el que se presenta en la Figura 14 que se muestra a continuación.

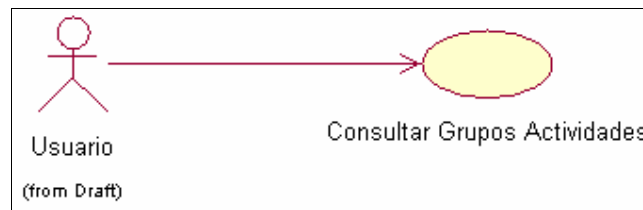


Figura 14. Diagrama de caso de uso: Gestión Usuarios-Actividades

- **Actores**

El actor del caso de uso es el usuario (personal de la Asociación) que se encuentra *logado* en la aplicación Web.

- **Descripción**

El presente caso de uso muestra la interacción que es necesario que el usuario lleve a cabo para poder realizar la matriculación o anulación de matrícula de un usuario (persona con Síndrome de Down) en un grupo de una actividad.

- **Precondiciones**

- El usuario ha seleccionado la opción *Usuarios-Actividades* del menú de la aplicación.

- **Postcondiciones**

- El usuario visualiza en la parte central de la pantalla un listado de todos los grupos creados para todas las actividades existentes en la aplicación.
- El usuario visualiza en la parte superior de la pantalla los campos por los que es posible realizar una búsqueda (actividad, curso académico, grupo y usuario), junto con el botón para llevar a cabo la citada búsqueda.

- **Flujo normal**

- El usuario accede a la opción *Usuarios-Actividades* del menú de la aplicación.
- El usuario visualiza, en la parte central de la pantalla, el listado actual de los grupos creados para todas las actividades existentes en la aplicación y, en la parte superior de la misma pantalla, la sección para poder realizar búsquedas sobre dichos grupos.
- El usuario selecciona la opción de realizar una búsqueda sobre los grupos de las actividades existentes en la aplicación a través del botón disponible para ello en la parte superior de la pantalla, habiendo introducido previamente, al menos, un criterio de búsqueda.
 - La aplicación mostrará en la parte central de la misma pantalla un listado mostrando la información de carácter general de todos aquellos grupos de actividades que satisfagan el criterio de búsqueda introducido.
 - Pulsando sobre la información de un grupo en concreto, la aplicación mostrará una nueva pantalla donde se visualizará, además de la información genérica del grupo, el listado de todos aquellos usuarios que se encuentran matriculados en el mismo.

- **Flujos alternativos**

- El servidor Web no se encuentra disponible y no es posible acceder a la aplicación.
- El usuario intenta acceder a la opción *Usuarios-Actividades* del menú para la que el perfil al que pertenece no tiene acceso.
- El usuario pulsa el botón para realizar una búsqueda sin haber introducido previamente, al menos, un criterio de búsqueda.
- El usuario introduce un criterio de búsqueda para el que no existen elementos en el sistema que lo satisfagan.

2.3.1.1.4.1.- Caso de Uso: Consultar Grupos-Actividades (Iteración 3)

La única opción existente, como se ha visto en el caso de uso anterior, dentro de la gestión de Usuarios–Actividades es la de poder realizar una consulta sobre los grupos existentes para cada una de las actividades ofertadas por la Asociación. A través de dicha consulta la aplicación permite que se lleve a cabo la matriculación de nuevos usuarios en las actividades y la anulación de las matriculas ya realizadas, así como la obtención del listado de todos los usuarios matriculados en un determinado grupo de una actividad. El caso de uso que se presenta en la Figura 15 responde a las acciones relativas a la gestión de Usuarios–Actividades que se acaban de mencionar.

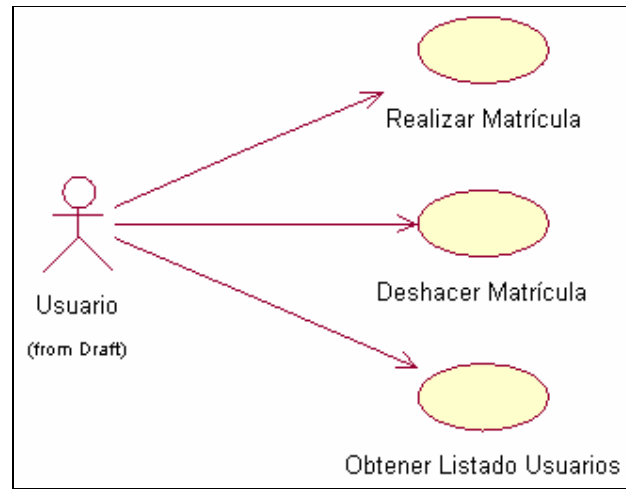


Figura 15. Diagrama de caso de uso: Consultar Grupos-Actividades

- **Actores**

El actor del caso de uso es el usuario (personal de la Asociación) que se encuentra *logado* en la aplicación Web.

- **Descripción**

El presente caso de uso muestra las posibles interacciones entre el usuario y el sistema para llevar a cabo matricula de un nuevo usuario en un grupo de una actividad o la anulación de una matricula ya existente.

- **Precondiciones**

- El usuario ha realizado de manera correcta una búsqueda sobre los grupos de una determinada actividad a través del buscador existente dentro de la opción *Usuarios-Actividades* del menú de la aplicación.

- **Postcondiciones**

- El usuario visualiza en la parte central de la pantalla un listado mostrando la información de carácter general de todos aquellos grupos de actividades que satisfacen el criterio de búsqueda introducido para realizar la consulta.

- **Flujo normal**

- Una vez localizado el grupo de la actividad deseado en el listado resultante de la búsqueda, el usuario podrá pulsar sobre la información de dicho grupo para acceder a una nueva pantalla donde se visualizará, además de la información genérica del grupo, un listado con todos los usuarios (personas con Síndrome de Down) matriculados en el mismo.

- Para cada uno de los usuarios matriculados actualmente en el grupo, la aplicación mostrará un botón para anular dicha matrícula. Al pulsar dicho botón la aplicación mostrará una nueva pantalla donde el usuario podrá modificar la fecha de anulación de dicha matrícula que, por defecto, será la fecha actual.

- Para cada uno de los usuarios matriculados en el grupo, pero que se encuentran dados de baja, la aplicación mostrará un botón para poder activar nuevamente la matrícula del usuario en el grupo. Al pulsar dicho botón la aplicación mostrará una nueva pantalla donde el usuario podrá modificar la fecha de reactivación de la matrícula que, por defecto, será la fecha actual.

La aplicación mostrará, junto al listado de usuarios matriculados en el grupo, un botón para permitir realizar nuevas matriculaciones. Al pulsar dicho botón se mostrará una nueva pantalla donde se seleccionará el usuario a matricular de entre todos los existentes y donde se podrá modificar la fecha de la matriculación que, por defecto, será la fecha actual.

- **Flujos alternativos**

- El servidor Web no se encuentra disponible y no es posible acceder a la aplicación.
- El usuario intenta dar de alta o de baja una matrícula o realizar una nueva matriculación cuando el perfil de usuario al que pertenece no tiene permisos de escritura sobre la opción *Usuarios-Actividades* del menú de la aplicación.
- El usuario intenta realizar una nueva matriculación de un usuario en un grupo cuando dicho usuario ya se encuentra dado de alta en el mismo grupo.
- El usuario, al dar nuevamente de alta un usuario que se encontraba dado de baja en un grupo de una actividad, introduce como fecha de alta una fecha anterior a la fecha en la que el usuario se había dado de baja.
- El usuario, al dar de baja un usuario que se encontraba dado de alta en un grupo de una actividad, introduce como fecha de baja una fecha anterior a la fecha en la que el usuario se había dado de alta.

2.3.1.1.5.- Caso de Uso: Gestión Entidades (Iteración 2)

La gestión de las entidades que colaboran con la Asociación es otra de las opciones disponibles desde el menú de la herramienta. A través de esta gestión, como se puede observar en el caso de uso mostrado en la Figura 16, el usuario podrá crear nuevas entidades colaboradoras o consultar y modificar la información de las que ya fueron creadas anteriormente.

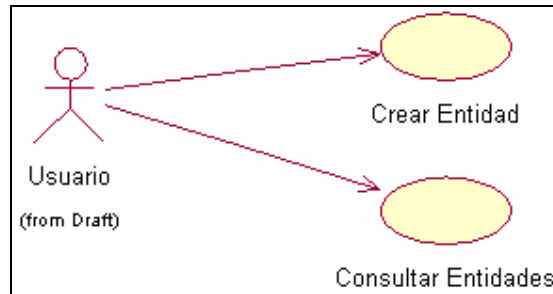


Figura 16. Diagrama de caso de uso: Gestión Entidades

Al observar el caso de uso mostrado en la figura anterior se puede comprobar que es idéntico al que se presentó en el apartado 2.3.1.1.1 cuando se trabajó sobre la gestión de las personas existentes en la Asociación. Por este motivo, en este momento, no se va a realizar una descripción detallada del caso de uso, sino que se va a explicar brevemente su flujo. Para obtener una descripción más detallada del caso de uso se remite al apartado 2.3.1.1.1 de la presente memoria donde se podrá encontrar toda la información correspondiente al caso de uso salvando las distancias entre la gestión de personas tratada en el citado apartado y la gestión de entidades que se está tratando actualmente.

Para llevar a cabo la gestión de entidades, el usuario accede a la opción *Entidades* del menú de la aplicación donde se le presentará un listado con todas las entidades que colaboran actualmente con la Asociación. En la parte superior de dicho listado dispondrá de una sección para realizar búsquedas así como de los botones que permiten realizar la búsqueda propiamente dicha y la creación de nuevas entidades. Si el usuario selecciona la opción de crear una nueva entidad la aplicación le presentará una nueva pantalla donde podrá introducir todos los datos relativos a la entidad que se desea crear. Entre la información a introducir para dar de alta una nueva entidad se encuentra la relativa a la dirección física de la entidad. El proceso para introducir dicha información queda reflejado en el siguiente caso de uso para cuya descripción se remite al apartado 2.3.1.1.1.1 de la memoria donde se explica el procedimiento a seguir para introducir una nueva dirección o utilizar una ya existente.

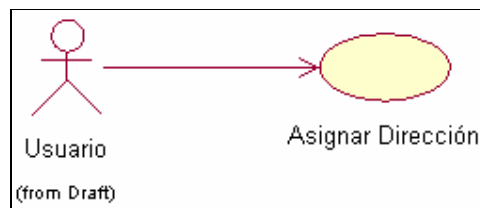


Figura 17. Diagrama de caso de uso: Crear Entidad

Si el usuario selecciona la opción de realizar una búsqueda sobre las entidades que ya colaboran con la Asociación deberá introducir el nombre de la entidad a buscar y, cuando lo haga, la aplicación mostrará la información relativa a dicha entidad permitiendo que el usuario modifique todos los datos mostrados (para obtener una descripción detallada del proceso de modificación de la dirección de una entidad se remite al apartado 2.3.1.1.1.2 de la memoria donde se describe el mismo proceso para la modificación de la dirección de una persona). Además, al realizar una búsqueda sobre una entidad, la aplicación mostrará el listado de subvenciones que han sido concedidas históricamente por dicha entidad permitiendo filtrar dicha información por el año de concesión. El caso de uso que describe los procesos a seguir para realizar una consulta sobre las entidades es el que se muestra en la Figura 18 presentada a continuación.

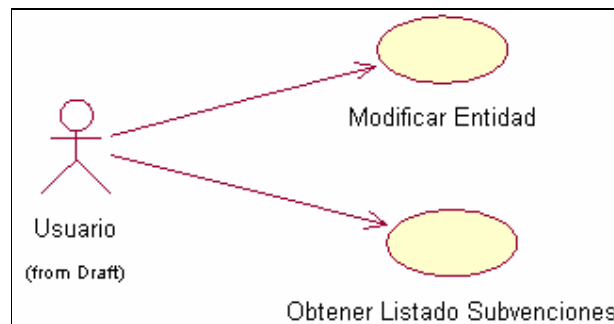


Figura 18. Diagrama de caso de uso: Consultar Entidades

2.3.1.1.6.- Caso de Uso: Gestión Subvenciones (Iteración 2)

Como ya se ha ido comentando a lo largo de la redacción de la presente memoria, otra de las gestiones que debe permitir realizar la herramienta a implementar es la correspondiente a la gestión de las subvenciones que le son concedidas a la Asociación Navarra de Síndrome de Down. A través de dicha gestión, como se puede observar en el caso de uso que se muestra en la Figura 19, se debe permitir la creación de nuevas subvenciones, así como la consulta de las ya existentes. Además, una vez realizada una consulta se deberá permitir la modificación de la información relativa a la subvención así como de las facturas y salarios presentados a la misma.

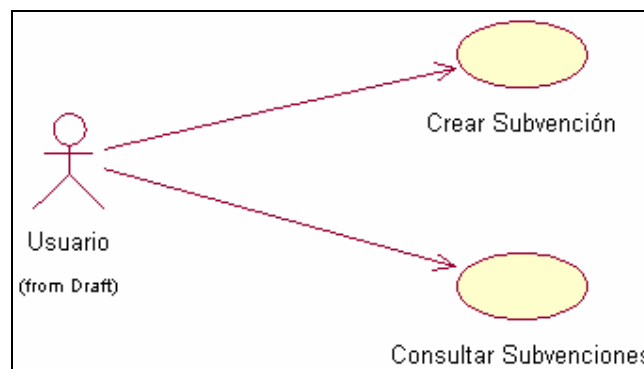


Figura 19. Diagrama de caso de uso: Gestión Subvenciones

- **Actores**

El actor del caso de uso es el usuario (personal de la Asociación) que se encuentra *logado* en la aplicación Web.

- **Descripción**

El presente caso de uso muestra las posibles interacciones entre el usuario y el sistema para llevar a cabo la gestión de las subvenciones. Esto es, muestra las interacciones que deberán realizarse para poder crear nuevas subvenciones o realizar consultas y modificaciones sobre las mismas.

- **Precondiciones**

- El usuario ha seleccionado la opción *Subvenciones* del menú de la aplicación.

- **Postcondiciones**

- El usuario visualiza en la parte central de la pantalla un listado de todas las subvenciones existentes en la aplicación.
- El usuario visualiza en la parte superior de la pantalla los campos por los que es posible realizar una búsqueda de subvenciones (nombre de la subvención, año de solicitud o entidad que la concede), además de la opción de crear nuevas subvenciones.

- **Flujo normal**

- El usuario accede a la opción *Subvenciones* del menú de la aplicación.
- El usuario visualiza, en la parte central de la pantalla, el listado actual de subvenciones existentes en la aplicación y, en la parte superior de la misma pantalla, la sección para poder realizar búsquedas sobre las subvenciones ya existentes en el sistema. Además, el usuario dispondrá de los botones para poder crear subvenciones y consultar las ya existentes.
- El usuario selecciona la opción de crear una nueva subvención a través del botón disponible para ello en la parte superior de la pantalla.
 - La aplicación mostrará una nueva pantalla donde introducir la información relativa a la subvención a crear (nombre, año, entidad que la concede, tipo de subvención, fecha de solicitud, cantidad presupuestada, cantidad solicitada, cantidad concedida y estado de la subvención).
- El usuario selecciona la opción de realizar una búsqueda sobre las subvenciones existentes en la aplicación a través del botón disponible para ello en la parte superior de la pantalla, habiendo introducido previamente, al menos, un criterio de búsqueda.
 - La aplicación mostrará en la parte central de la misma pantalla un listado mostrando la información de todas aquellas subvenciones que satisfagan el criterio de búsqueda introducido.

- Pulsando sobre la información de una subvención en concreto, la aplicación mostrará una nueva pantalla donde se visualizará, además de la información genérica de la subvención, el listado de facturas y salarios presentados a la misma, así como la posibilidad de asignar nuevas facturas y/o salarios a la citada subvención.
- **Flujos alternativos**
 - El servidor Web no se encuentra disponible y no es posible acceder a la aplicación.
 - El usuario intenta acceder a la opción *Subvenciones* del menú para la que el perfil al que pertenece no tiene acceso.
 - El usuario intenta crear una nueva subvención cuando el perfil de usuario al que pertenece no tiene permisos de escritura sobre la opción *Subvenciones* del menú de la aplicación.
 - El usuario no introduce correctamente todos los datos necesarios para poder crear una nueva subvención.
 - El usuario pulsa el botón para realizar una búsqueda sin haber introducido previamente, al menos, un criterio de búsqueda.
 - El usuario introduce un criterio de búsqueda para el que no existen subvenciones en la aplicación.

2.3.1.1.6.1.- Caso de Uso: Consultar Subvenciones (Iteración 3)

En la presentación del caso de uso anterior relativo a la gestión de subvenciones se ha indicado que, una vez realizada una consulta sobre las subvenciones, la aplicación debe permitir realizar modificaciones sobre las subvenciones consultadas, así como sobre las facturas y salarios presentadas a las mismas. Asimismo, una vez realizada una consulta sobre las subvenciones concedidas a la Asociación, la aplicación debe permitir la asignación de nuevas facturas y/o salarios a la subvención en cuestión. El caso de uso presentado en la siguiente figura representa el procedimiento a seguir para llevar a cabo las acciones que se acaban de mencionar como posibles después de realizar una búsqueda sobre las subvenciones concedidas a la Asociación.

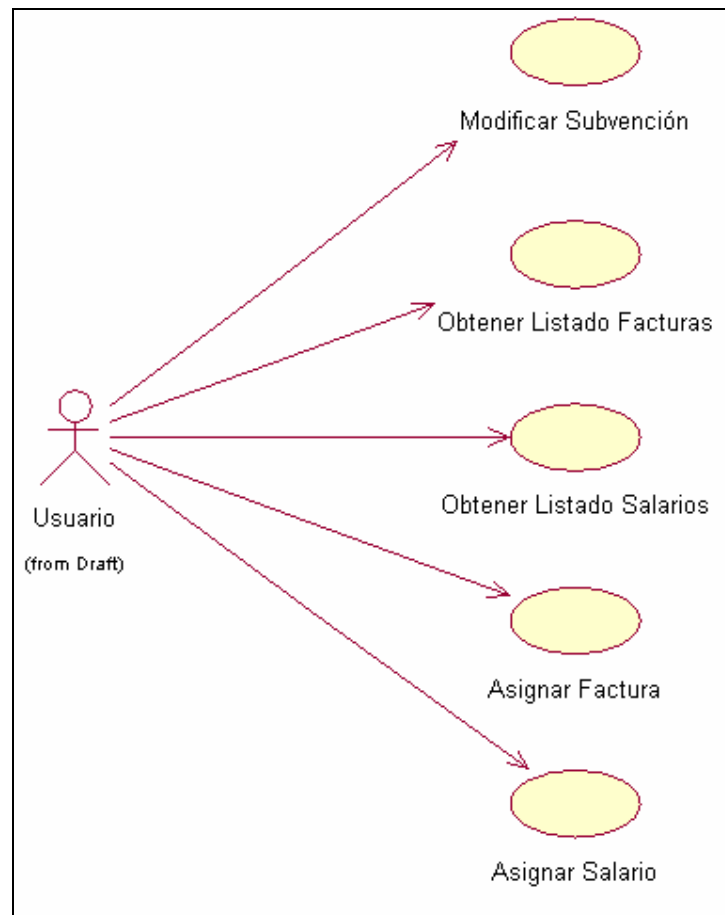


Figura 20. Diagrama de caso de uso: Consultar Subvenciones

- **Actores**

El actor del caso de uso es el usuario (personal de la Asociación) que se encuentra *logado* en la aplicación Web.

- **Descripción**

El presente caso de uso muestra las posibles interacciones entre el usuario y el sistema para llevar a cabo la modificación de una subvención, de las facturas o salarios presentados a la misma o para realizar la asignación de una nueva factura o salario a la subvención en cuestión.

- **Precondiciones**

- El usuario ha realizado de manera correcta una búsqueda de subvenciones a través del buscador existente dentro de la opción *Subvenciones* del menú de la aplicación.

- **Postcondiciones**

- El usuario visualiza en la parte central de la pantalla un listado mostrando la información de todas aquellas subvenciones que satisfacen el criterio de búsqueda introducido para realizar la consulta.

- **Flujo normal**

- Una vez localizada la subvención deseada en el listado resultante de la búsqueda, el usuario podrá pulsar sobre la información de dicha subvención para acceder a una nueva pantalla donde se visualizará, además de la información genérica de la subvención, el listado de facturas y salarios presentados a la citada subvención y donde se podrán realizar las modificaciones que se consideren oportunas. Asimismo, la aplicación mostrará junto a los listados de facturas y salarios la suma total de los importes de las citadas facturas y salarios que se han presentado a la subvención, así como un campo que indique la cantidad que resta para alcanzar el total del importe concedido por la subvención.

- Si el usuario pulsa el botón para asignar una nueva factura a la subvención seleccionada, la aplicación mostrará una nueva pantalla donde se deberá elegir la factura deseada de entre el listado total de facturas existentes. Una vez seleccionada la factura, se deberá indicar el porcentaje del importe de dicha factura que se presenta a la subvención y la aplicación deberá calcular de manera automática la cantidad que supone dicho porcentaje. Del mismo modo, al seleccionar una de las facturas existentes, la aplicación debe indicar como fecha de asignación la fecha actual permitiendo al usuario modificar dicha fecha.

Este mismo procedimiento deberá seguirse para el caso en el que se desee asignar el salario de un profesional a una determinada subvención. Es decir, deberá pulsarse el botón “Asignar salario” existente para ello, seleccionar el profesional cuyo salario se desea asignar e indicar el porcentaje de dicho salario que se va a presentar a la subvención. Al igual que en el caso de las facturas, la fecha de asignación será la fecha actual, fecha que será posible modificar.

- Si el usuario desea modificar la información de una de las facturas asignadas a la subvención deberá pulsar sobre el registro correspondiente a dicha factura en el listado de facturas presentadas a la subvención. Al hacerlo, la aplicación mostrará una nueva pantalla con el detalle de dicha factura permitiendo al usuario modificar el porcentaje del importe de la factura que se presenta a la subvención. Al realizar la citada modificación, la aplicación actualizará la fecha de asignación de la factura a la fecha actual, permitiendo al usuario modificar dicha fecha.

De la misma manera que en el caso de la asignación de salarios, el procedimiento para realizar la modificación de un salario asignado a una subvención es idéntico al presentado para realizar una modificación sobre una factura presentada a una subvención.

- **Flujos alternativos**

- El servidor Web no se encuentra disponible y no es posible acceder a la aplicación.
- El usuario intenta asignar la misma factura más de una vez a la misma subvención.
- El usuario intenta asignar el salario del mismo profesional más de una vez a la misma subvención.
- El usuario, al asignar una factura a una subvención o al modificar la cantidad ya asignada, indica como importe de asignación una cantidad que, sumada a las cantidades de la misma factura que ya han sido presentadas a otras subvenciones, supera la cantidad total de la factura.
- El usuario, al asignar un salario de un profesional a una subvención o al modificar la cantidad ya asignada, indica como importe de asignación una cantidad que, sumada a las cantidades del salario del mismo profesional que ya han sido presentadas a otras subvenciones, supera el salario anual del citado profesional.

2.3.1.1.7.- Caso de Uso: Gestión Proveedores (Iteración 2)

En el presente subapartado correspondiente a la realización de los casos de uso que describen la funcionalidad que debe aportar la herramienta a implementar se debería describir el caso de uso correspondiente a la gestión de los proveedores con los que trabaja la Asociación. Sin embargo, realizando una revisión sobre los requisitos exigidos por la Asociación para poder llevar a cabo la citada gestión de proveedores se observa que solicitan la posibilidad de dar de alta nuevos proveedores, la posibilidad de realizar consultas y modificaciones sobre los proveedores ya existentes, así como poder obtener un listado de las facturas generadas por sus proveedores (los casos de uso correspondientes a los citados requerimientos se muestran en las dos figuras presentadas a continuación) requisitos que, como acabamos de observar en la realización de los casos de uso correspondientes a la gestión de subvenciones, coinciden en la mayor parte con los allí presentados. Por este motivo, se remite al apartado anterior para obtener una descripción detallada del procedimiento a seguir para llevar a cabo la gestión de proveedores.

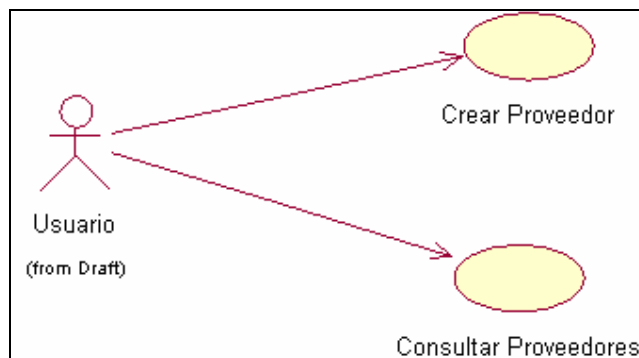


Figura 21. Diagrama de caso de uso: Gestión Proveedores

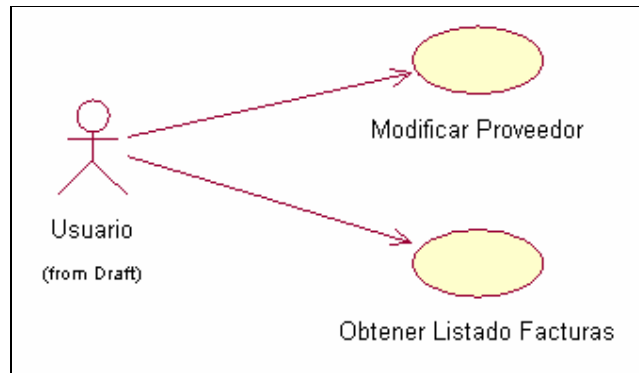


Figura 22. Diagrama de caso de uso: Consultar Proveedores

2.3.1.1.8.- Caso de Uso: Gestión Facturas (Iteración 2)

Al igual que en el resto de casos de uso presentados a lo largo de esta sección de la memoria, los requisitos exigidos por parte de la Asociación Navarra de Síndrome de Down para poder llevar a cabo la gestión de las facturas se centraban en el hecho de poder insertar nuevas facturas y de poder realizar consultas y modificaciones sobre las facturas ya introducidas en el sistema. Los casos de uso correspondientes a las acciones citadas se muestran en las figuras que se presentan a continuación.

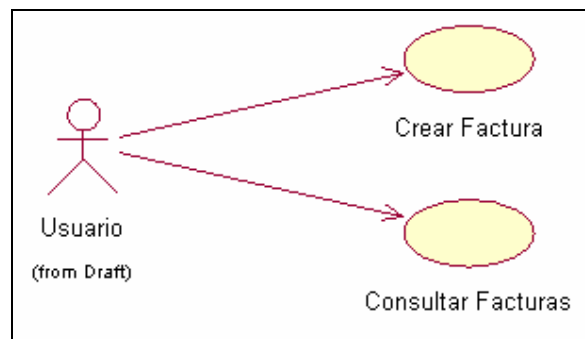


Figura 23. Diagrama de caso de uso: Gestión Facturas

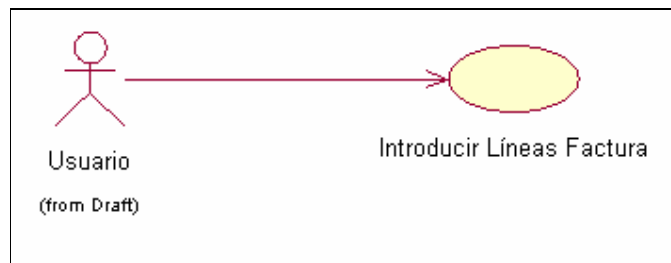


Figura 24. Diagrama de caso de uso: Crear Factura



Figura 25. Diagrama de caso de uso: Consultar Facturas

Como se puede comprobar al observar las anteriores figuras, la manera de proceder para realizar la gestión de facturas es idéntica a la que ya se ha venido describiendo en el resto de casos de uso por lo que se remite a cualquiera de ellos para obtener una explicación sobre la manera de proceder, resaltando en este momento como aspecto diferenciador el hecho de que, para poder realizar la inserción de una nueva factura, el usuario dispondrá de una pantalla donde introducir la información de carácter general de la factura (ID, proveedor, fecha, concepto, importe, etc.) y de una sección desde la cual, a través de un botón, se accederá a una nueva pantalla en la que se podrán ir introduciendo todas y cada una de las líneas que constituyan la nueva factura.

2.3.1.1.9.- Caso de Uso: Gestión Salarios (Iteración 2)

Relacionado con la gestión de profesionales que se ha descrito dentro del apartado correspondiente a la gestión de personas, el personal de la Asociación Navarra de Síndrome de Down solicitó como requisito el hecho de que la aplicación a implementar permitiera llevar a cabo una gestión de los salarios recibidos por cada profesional. En este sentido, tal y como se observa en las Figuras 26 y 27 que se presentan a continuación, se solicitaba que la aplicación permitiera la introducción de nuevos salarios para cada profesional así como realizar consultas y modificaciones sobre los salarios ya introducidos en el sistema.

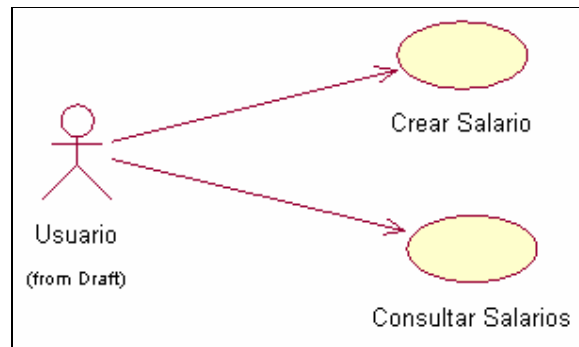


Figura 26. Diagrama de caso de uso: Gestión Salarios

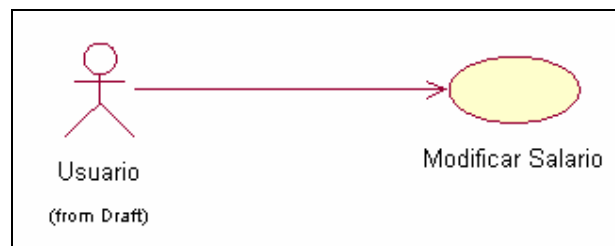


Figura 27. Diagrama de caso de uso: Consultar Salarios

Como ya viene ocurriendo en los últimos casos de uso presentados, el flujo de realización de las acciones solicitadas por parte de la Asociación para poder llevar a cabo la gestión de salarios coincide con los que se han venido realizando a lo largo de todo el apartado. Es decir, al acceder a la sección *Salarios* del menú de la aplicación el usuario se encontrará con un listado de los salarios correspondientes al año actual existentes en la aplicación, así como con un botón para poder insertar nuevos salarios y una sección en la que poder realizar búsquedas de salarios por profesional mes y año. En el caso de que se seleccione la opción de insertar un nuevo salario la aplicación presentará una nueva pantalla donde el usuario podrá introducir la información

necesaria para poder llevar a cabo la inserción. Por el contrario, si se realiza una búsqueda sobre los salarios existentes, la aplicación mostrará en la misma pantalla aquellos registros que coincidan con la búsqueda realizada y permitirá que, pulsando sobre cada uno de ellos, se acceda a una nueva pantalla donde visualizar en detalle los datos del salario y donde poder realizar modificaciones sobre los mismos siempre y cuando el salario en cuestión no haya sido presentado a ninguna de las subvenciones concedidas a la Asociación.

2.3.1.1.10.- Caso de Uso: Gestión Gastos Extraordinarios (Iteración 2)

La penúltima de las opciones del menú de la aplicación que ofrece funcionalidad relacionada directamente con la actividad llevada a cabo por la Asociación Navarra de Síndrome de Down es la gestión de gastos extraordinarios realizados por los profesionales y/o voluntarios del centro cuyos casos de uso se presentan en esta sección.

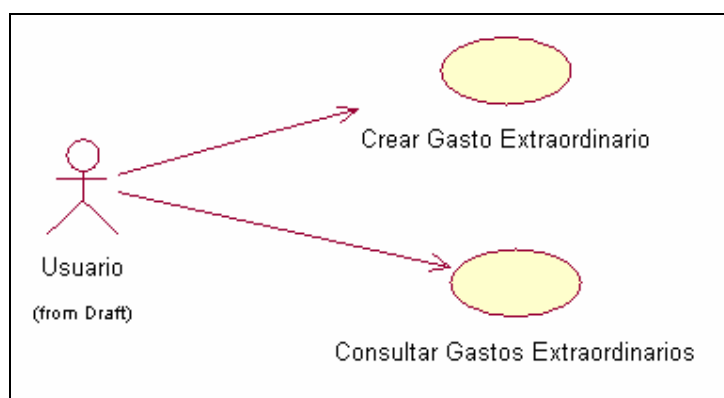


Figura 28. Diagrama de caso de uso: Gestión Gastos Extraordinarios

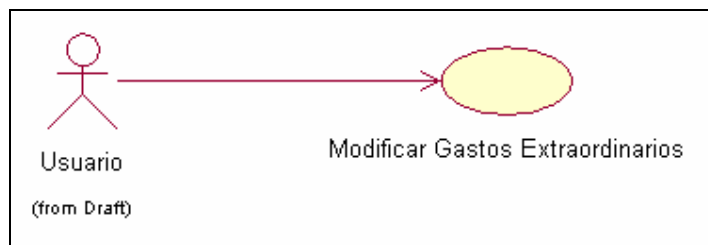


Figura 29. Diagrama de caso de uso: Consultar Gastos Extraordinarios

En esta ocasión, y debido a que los casos de uso presentados coinciden con los que ya han sido estudiados para otras de las opciones del menú de la aplicación, no se va a realizar un análisis detallado de los mismos sino que se va a presentar de manera genérica el procedimiento que seguirá la aplicación para llevar a cabo la gestión de los gastos extraordinarios de los profesionales o voluntarios. Una vez que el usuario acceda a la sección *Gastos Extraordinarios* del menú de la aplicación ésta presentará un listado con todos los gastos realizados por los profesionales y/o voluntarios así como la posibilidad de insertar nuevos gastos y de realizar búsquedas entre los ya existentes, filtrando a través de los campos relativos al tipo de persona (profesional o voluntario), al nombre de la persona y al concepto del gasto. Pulsando sobre el botón que permite insertar un nuevo gasto la aplicación mostrará una nueva pantalla donde el usuario deberá introducir la información necesaria para llevar a cabo la inserción. En este momento es importante indicar que la aplicación obligará al usuario a introducir el tipo

de persona al que se desea vincular el gasto ya que, en función de este dato, la aplicación mostrará el listado de profesionales o voluntarios existentes en el sistema para poder asociarles el gasto que se está introduciendo. Si por el contrario, desde la pantalla inicial del menú *Gastos Extraordinarios*, se realiza una búsqueda sobre los gastos existentes en el sistema, la aplicación mostrará un listado con todos aquellos registros que cumplan el criterio de búsqueda introducido y permitirá acceder al detalle de todos y cada uno de ellos bien para realizar una consulta o bien para realizar una modificación de la información mostrada.

2.3.1.1.11.- Caso de Uso: Gestión Actividades Extraordinarias (Iteración 2)

Para finalizar con las opciones disponibles en el menú de la aplicación que ofrecen funcionalidad directamente relacionada con la actividad llevada a cabo por la Asociación Navarra de Síndrome de Down, en esta sección se van a presentar los casos de uso correspondientes a la gestión de las actividades que pueden realizar los profesionales de la Asociación fuera de la actividad específica para la que han sido contratados.

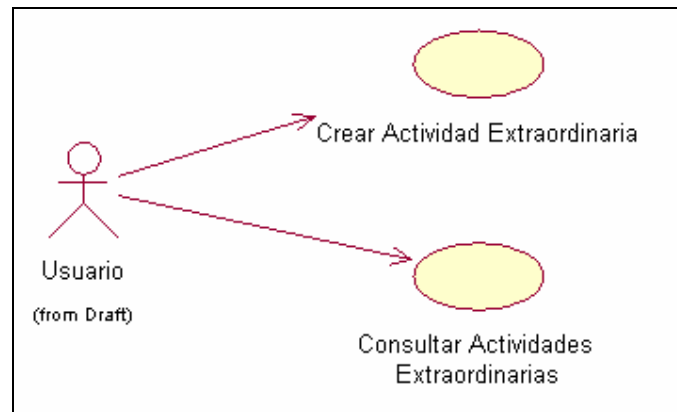


Figura 30. Diagrama de caso de uso: Gestión Actividades Extraordinarias

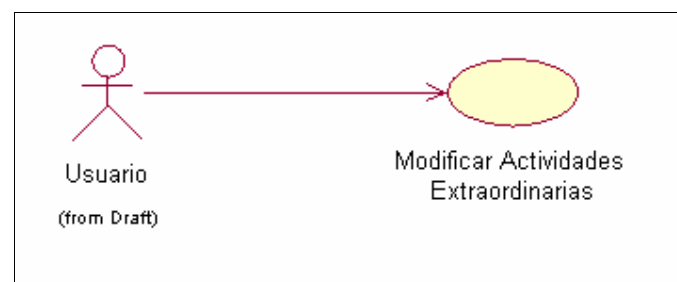


Figura 31. Diagrama de caso de uso: Consultar Actividades Extraordinarias

Al observar los casos de uso presentados en las figuras 30 y 31 se observa que coinciden exactamente con los que se han tratado en el apartado 2.3.1.1.10 del presente documento relativo a la gestión de los gastos extraordinarios de profesionales y voluntarios. Por este motivo, se remite a dicho apartado para obtener una explicación del procedimiento que debe seguir el usuario para poder realizar tanto inserciones como consultas y modificaciones de la información correspondiente a las actividades extraordinarias realizadas por los profesionales dentro de la Asociación. Se debe indicar en este momento, como diferencias reseñables entre los casos de uso actuales y los presentados en el apartado 2.3.1.1.10, que en este caso, las búsquedas se podrán realizar

filtrando por tipo de actividad realizada, curso en el que se realiza o profesional que la lleva a cabo. Asimismo se debe indicar que la aplicación mostrará un error siempre que el usuario intente insertar dos registros diferentes para el mismo profesional y la misma actividad en el mismo curso académico.

2.3.1.1.12.- Caso de Uso: Gestión Perfiles (Iteración 2)

Todos los casos de uso presentados hasta el momento responden a opciones del menú que permiten dotar a la aplicación de la funcionalidad requerida por la Asociación Navarra de Síndrome de Down. Además de las citadas opciones del menú, la aplicación cuenta con otras opciones adicionales que permitirán al personal administrativo de la Asociación controlar los usuarios que acceden a la herramienta así como los permisos de acceso sobre la misma. El caso de uso que se muestra en la Figura 32 corresponde a la opción del menú a través de la cual el personal de la Asociación podrá controlar los diferentes perfiles de acceso a la herramienta que podrán tener los usuarios de la misma.

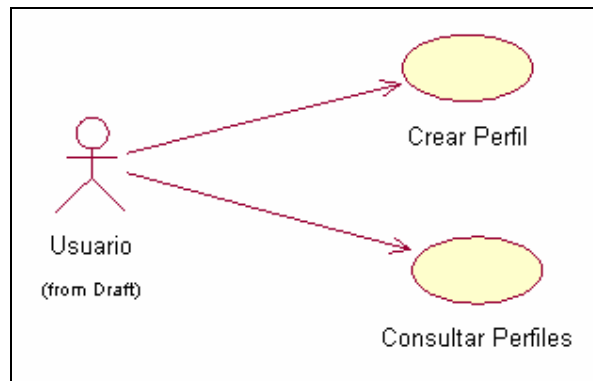


Figura 32. Diagrama de caso de uso: Gestión Perfiles

- **Actores**
El actor del caso de uso es el usuario (personal de la Asociación) que se encuentra *logado* en la aplicación Web.
- **Descripción**
El presente caso de uso muestra las posibles interacciones entre el usuario y el sistema para llevar a cabo la gestión de los diferentes perfiles que podrán tener los usuarios que empleen la herramienta. Es decir, a través del caso de uso se pretende describir la manera de proceder para poder definir nuevos perfiles de usuario y modificar o borrar los ya existentes.
- **Precondiciones**
 - El usuario ha seleccionado la opción *Perfiles* del menú de la aplicación.
- **Postcondiciones**
 - El usuario visualiza en la parte central de la pantalla un listado de todos los perfiles existentes en ese momento en la aplicación, así como las opciones que permiten realizar una consulta sobre los perfiles existentes o crear un perfil de usuario nuevo.

- **Flujo normal**

- El usuario accede a la opción *Perfiles* del menú de la aplicación.
- El usuario visualiza, en la parte central de la pantalla, el listado actual de perfiles existentes en la aplicación y, en la parte superior de la misma pantalla, la sección para poder realizar búsquedas sobre los perfiles ya existentes en el sistema. Además, el usuario dispondrá de los botones para poder crear nuevos perfiles y consultar los ya existentes.
- El usuario selecciona la opción de crear un nuevo perfil de usuario a través del botón disponible para ello en la parte superior de la pantalla.
 - La aplicación mostrará una nueva pantalla donde se deberá introducir el nombre que se desea dar el perfil de usuario en creación.
 - En la misma pantalla la aplicación mostrará una estructura tabular donde, para cada una de las opciones disponibles en el menú de la herramienta, se presentarán las tres opciones de acceso posibles (sin acceso, lectura y escritura). Por defecto, la aplicación indicará que el perfil a crear no tendrá acceso a ninguna de las opciones existentes en el menú, con la excepción de la opción de *Cambio Password* para la que se mostrará por defecto acceso de escritura. El usuario, para llevar a cabo la creación del nuevo perfil de usuario, podrá modificar a su antojo los tipos de acceso que se desea que tenga el nuevo perfil sobre cada opción del menú. Se debe indicar que, para el caso de las opciones del menú correspondientes a *Perfiles* y *Personal*, no será posible modificar el tipo de acceso (no será posible seleccionar las opciones de lectura o escritura) puesto que el único perfil de usuario con acceso a dichas opciones del menú será el de “Superadministrador”, perfil que vendrá creado por defecto en la herramienta.
- El usuario selecciona la opción de realizar una búsqueda sobre los perfiles de usuario existentes en la aplicación a través del botón disponible para ello en la parte superior de la pantalla, habiendo introducido previamente el criterio de búsqueda necesario para que la búsqueda sea efectiva.
 - La aplicación mostrará en la parte central de la misma pantalla un listado mostrando el nombre de todos aquellos perfiles de usuario existentes en la aplicación que satisfagan el criterio de búsqueda introducido.
 - Pulsando sobre el nombre de un perfil de usuario en concreto, la aplicación mostrará una nueva pantalla donde se visualizarán los distintos tipos de acceso que otorga el perfil en cuestión a cada una de las opciones del menú de la aplicación.

- **Flujos alternativos**

- El servidor Web no se encuentra disponible y no es posible acceder a la aplicación.
- El usuario intenta acceder a la opción *Perfiles* del menú para la que el perfil al que pertenece no tiene acceso.
- El usuario intenta crear un nuevo perfil de usuario asignándole un nombre ya existente para otro de los perfiles creados en el sistema.
- El usuario pulsa el botón para realizar una búsqueda sin haber introducido previamente, al menos, un criterio de búsqueda.
- El usuario introduce un criterio de búsqueda para el que no existen perfiles de usuario en la aplicación.

2.3.1.1.12.1.- Caso de Uso: Consultar Perfiles (Iteración 3)

Además de la creación de nuevos perfiles de usuario que ya se ha visto en el apartado anterior, para realizar una correcta gestión de los distintos perfiles de que consta la aplicación, es necesario que la herramienta permita borrar los perfiles ya obsoletos o realizar modificaciones en los ya existentes, tal y como se describe en el caso de uso de la Figura 33.

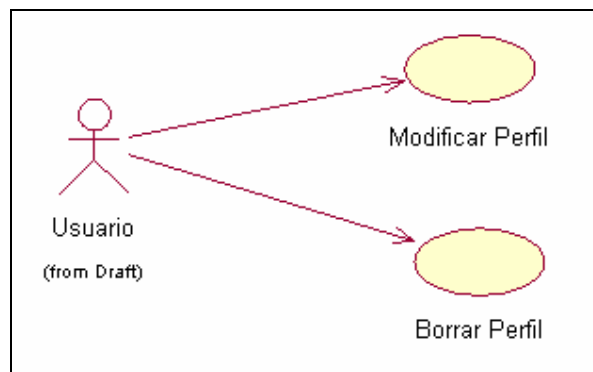


Figura 33. Diagrama de caso de uso: Consultar Perfiles

- **Actores**

El actor del caso de uso es el usuario (personal de la Asociación) que se encuentra *logado* en la aplicación Web.

- **Descripción**

El presente caso de uso muestra las posibles interacciones entre el usuario y el sistema para llevar a cabo la modificación de los accesos a las opciones del menú de un perfil determinado o para eliminar de la aplicación un perfil de usuario ya en desuso.

- **Precondiciones**

- El usuario ha realizado de manera correcta una búsqueda de perfiles a través del buscador existente dentro de la opción *Perfiles* del menú de la aplicación.

- **Postcondiciones**
 - El usuario visualiza en la parte central de la pantalla un listado mostrando los registros correspondientes a los perfiles que coincidan con el nombre introducido por el usuario al realizar la búsqueda.
- **Flujo normal**
 - Una vez localizado el perfil deseado en el listado resultante de la búsqueda, el usuario podrá pulsar sobre el nombre de dicho perfil para acceder a una nueva pantalla donde se visualizarán todos los accesos de los que consta el perfil seleccionado así como los botones de grabar y borrar para hacer efectivas las acciones de modificación o borrado.
 - Si el usuario *logado* en el sistema desea realizar una modificación sobre los accesos del perfil en cuestión deberá seleccionar el nuevo tipo de acceso que se desee dar a cada una de las opciones del menú y, una vez realizadas todas las modificaciones pertinentes, pulsar el botón grabar para que los cambios realizados sean efectivos.
 - Si el usuario *logado* en el sistema desea eliminar el perfil al que se ha accedido a través de la búsqueda deberá pulsar el botón borrar existente para ello en la pantalla de detalle del perfil.
- **Flujos alternativos**
 - El servidor Web no se encuentra disponible y no es posible acceder a la aplicación.
 - El usuario intenta borrar un perfil de usuario para el que hay usuarios asignados.

2.3.1.1.13.- Caso de Uso: Gestión Personal (Iteración 2)

Como ya se ha comentado al presentar los dos casos de uso anteriores, la aplicación además de contar con las opciones del menú relacionadas directamente con la funcionalidad que se espera que proporcione la herramienta, cuenta con opciones a través de las cuales será posible gestionar tanto los perfiles de usuario como los usuarios propiamente dichos que podrán acceder a la herramienta. Los diagramas de caso de uso relacionados con la gestión de perfiles se han presentado en los dos apartados anteriores por lo que, en este momento, se va a realizar la presentación de los diagramas correspondientes a la gestión de usuarios que, como se observa en la Figura 34, pasa por permitir tanto la creación como la consulta de los usuarios existentes (como ya viene siendo habitual, a través de la consulta de usuarios se permitirá la realización de modificaciones y borrado de usuarios).

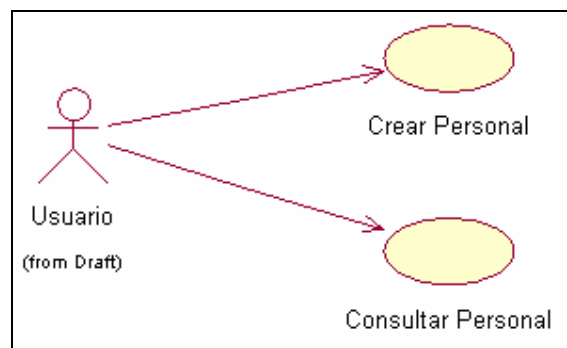


Figura 34. Diagrama de caso de uso: Gestión Personal

- **Actores**
 - El actor del caso de uso es el usuario (personal de la Asociación) que se encuentra *logado* en la aplicación Web.
- **Descripción**
 - A través del caso de uso mostrado en la Figura 34 se pretende hacer ver el procedimiento que deberá seguir el usuario superadministrador del sistema para poder crear nuevos usuarios o para modificar y borrar los ya existentes.
- **Precondiciones**
 - El usuario ha seleccionado la opción *Personal* del menú de la aplicación.
- **Postcondiciones**
 - El usuario visualiza en la parte central de la pantalla un listado de todos los usuarios existentes en la aplicación, así como las opciones de consulta y creación de usuarios.

- **Flujo normal**

- El usuario accede a la opción *Personal* del menú de la aplicación.
- El usuario visualiza, en la parte central de la pantalla, el listado actual de usuarios existentes en la aplicación y, en la parte superior de la misma pantalla, la sección para poder realizar búsquedas sobre los usuarios ya existentes en el sistema. Además, el usuario dispondrá de los botones para poder crear nuevos usuarios y consultar los ya existentes.
- El usuario selecciona la opción de crear un nuevo usuario a través del botón disponible para ello en la parte superior de la pantalla.
 - La aplicación mostrará una nueva pantalla donde se deberá introducir la información necesaria para la creación del nuevo usuario. En concreto se deberá indicar el nombre del nuevo usuario, su *login* y *password* de acceso a la herramienta así como el perfil de usuario al que se va a asociar el usuario que se está creando (la asignación del perfil de usuario se realizará a través de un combo de selección donde se presentarán todos los perfiles de usuario existentes en el sistema).
- El usuario selecciona la opción de realizar una búsqueda sobre los usuarios existentes en la aplicación a través del botón disponible para ello en la parte superior de la pantalla, habiendo introducido previamente uno de los posibles criterios de búsqueda necesario para que la búsqueda sea efectiva (nombre de usuario, *login* o perfil de usuario).
 - La aplicación mostrará en la parte central de la misma pantalla un listado mostrando la información de todos aquellos usuarios existentes en la aplicación que satisfagan el criterio de búsqueda introducido.
 - Pulsando sobre la información de un usuario en concreto, la aplicación mostrará una nueva pantalla donde se visualizará en detalle la información del usuario y donde será posible modificarla o borrar el usuario en cuestión.

- **Flujos alternativos**

- El servidor Web no se encuentra disponible y no es posible acceder a la aplicación.
- El usuario intenta acceder a la opción *Personal* del menú para la que el perfil al que pertenece no tiene acceso.
- El usuario intenta crear un nuevo usuario asignándole un *login* ya existente para otro de los usuarios existentes en el sistema.
- El usuario intenta asignar al nuevo usuario el perfil de “Superadministrador”.
- El usuario pulsa el botón para realizar una búsqueda sin haber introducido previamente, al menos, un criterio de búsqueda.
- El usuario introduce un criterio de búsqueda para el que no existen usuarios en la aplicación.

2.3.1.1.13.1.- Caso de Uso: Consultar Personal (Iteración 3)

Como ya se ha indicado anteriormente, para realizar una correcta gestión del personal de la Asociación Navarra de Síndrome de Down que podrá hacer uso de la aplicación es necesario, además de poder crear nuevos usuarios para la herramienta, poder realizar modificaciones sobre la información de los usuarios ya creados o eliminar aquellos usuarios a los que ya no se desea permitir el acceso a la aplicación. El diagrama de casos de uso correspondiente tanto a la modificación como al borrado de usuarios es el que nos ocupa en este momento y se muestra en la Figura 35.

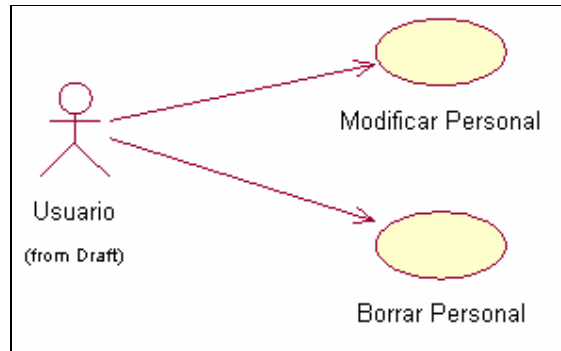


Figura 35. Diagrama de caso de uso: Consultar Personal

- **Actores**
El actor del caso de uso es el usuario (personal de la Asociación) que se encuentra *logado* en la aplicación Web.
- **Descripción**
El presente caso de uso muestra las posibles interacciones entre el usuario y el sistema para llevar a cabo el borrado o la modificación de la información de los usuarios que acceden a la aplicación.
- **Precondiciones**
 - El usuario ha realizado de manera correcta una búsqueda de usuarios a través del buscador existente dentro de la opción *Personal* del menú de la aplicación.
- **Postcondiciones**
 - El usuario visualiza en la parte central de la pantalla un listado mostrando los registros correspondientes a los usuarios que coincidan con la búsqueda realizada.

- **Flujo normal**

- Una vez localizado el usuario deseado en el listado resultante de la búsqueda, el usuario superadministrador podrá pulsar sobre la información de dicho usuario para acceder a una nueva pantalla donde se podrá llevar a cabo la modificación o borrado del usuario en cuestión.
 - Si el usuario superadministrador desea realizar una modificación sobre la información del usuario deberá pulsar el botón grabar existente en la pantalla después de haber realizado todos los cambios que se consideren necesarios.
 - Si el usuario superadministrador desea eliminar el usuario al que se ha accedido a través de la búsqueda deberá pulsar el botón borrar existente para ello.

- **Flujos alternativos**

- El servidor Web no se encuentra disponible y no es posible acceder a la aplicación.
- El usuario intenta borrar el usuario asignado al perfil de “Superadministrador”.
- El usuario modifica la información del usuario asignándole un *login* ya existente en el sistema.
- El usuario modifica la información del usuario asignándole el perfil de “Superadministrador”.

2.3.1.1.14.- Caso de Uso: Gestión Cambio Password (Iteración 2)

Para finalizar con la presentación de los diagramas de casos de uso que describen la funcionalidad de la aplicación desarrollada para la Asociación Navarra de Síndrome de Down en este apartado se va a presentar el caso de uso correspondiente a la gestión del cambio de password que, aunque no es un caso relativo a la funcionalidad específica solicitada por el personal de la Asociación, se considera necesario para permitir que sean los propios usuarios quienes controlen la clave con la acceden a la herramienta.

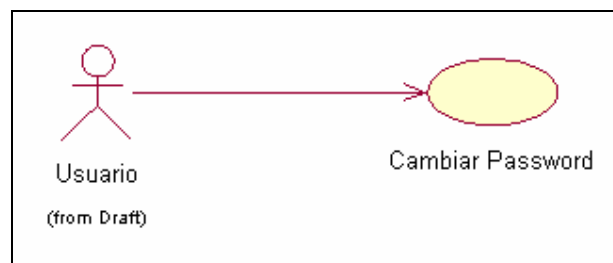


Figura 36. Diagrama de caso de uso: Gestión Cambio Password

- **Actores**
El actor del caso de uso es el usuario (personal de la Asociación) que se encuentra *logado* en la aplicación Web.
- **Descripción**
El presente caso de uso muestra las interacciones que el usuario debe realizar con el sistema para poder cambiar la contraseña que le da acceso a la aplicación.
- **Precondiciones**
 - El usuario ha seleccionado la opción *Cambio Password* del menú de la aplicación.
- **Postcondiciones**
 - El usuario visualiza en la parte central de la pantalla el listado de campos que debe cumplimentar para que el cambio de password tenga efecto.
- **Flujo normal**
 - El usuario accede a la opción *Cambio Password* del menú de la aplicación.
 - El usuario visualiza, en la parte central de la pantalla, el nombre de usuario con el que ha accedido a la aplicación así como los campos en los que deberá introducir tanto su clave de acceso antigua como la nueva clave que se desea asignar y la confirmación de la nueva contraseña introducida.
 - El usuario introduce la información solicitada por la aplicación (password antigua, password nueva y confirmación del nuevo password) obteniendo como resultado la correcta actualización de la clave de acceso a la herramienta.
- **Flujos alternativos**
 - El servidor Web no se encuentra disponible y no es posible acceder a la aplicación.
 - El usuario intenta acceder a la opción *Cambio Password* del menú para la que el perfil al que pertenece no tiene acceso.
 - El usuario intenta modificar su clave de acceso a la herramienta cuando el perfil de usuario al que pertenece no tiene permisos de escritura sobre la opción *Cambio Password* del menú de la aplicación.
 - El usuario no introduce todos los campos solicitados por la aplicación para poder realizar el cambio de contraseña.
 - El usuario introduce como password antigua una clave que no coincide con la que se encuentra asignada a su usuario.
 - El usuario introduce como confirmación de la nueva password un valor que no coincide con el introducido en el campo correspondiente a la nueva contraseña que se desea asignar.

2.4.- Diagrama Entidad-Relación

Para finalizar con el capítulo dedicado al análisis de requisitos de la aplicación a desarrollar, se va a presentar a continuación el diagrama Entidad-Relación que modela la Base de Datos que será necesario implementar para poder gestionar toda la información manejada por la Asociación Navarra de Síndrome de Down. Comenzaremos el apartado realizando una breve introducción al modelo Entidad-Relación para presentar y explicar finalmente el diagrama que será necesario implementar en una Base de Datos para dar cabida a todos los requisitos funcionales identificados en el apartado 2.2.3 del presente capítulo de la memoria descriptiva del proyecto.

El diagrama Entidad-Relación es una representación conceptual de la información empleada para el modelado de Bases de Datos que considera los objetos que pertenecen a la Base de Datos como entidades que contienen atributos y se vinculan entre ellas a través de relaciones [Def.DiagramaER]. Igualmente, de manera formal podemos definir un diagrama Entidad-Relación como un lenguaje gráfico para describir conceptos mientras que informalmente se podría considerar como simples dibujos o gráficos que describen la información que trata un sistema de información [Def.DiagramaER.II]. Los elementos que conforman un diagrama Entidad-Relación se describen a continuación por orden de importancia [ElementosER].

- **Entidad**

Una entidad es cualquier objeto o concepto del mundo real sobre el que se tiene información. Existen dos tipos de entidades: fuertes y débiles. Una entidad débil es aquella cuya existencia depende de la existencia de otra entidad mientras que una entidad fuerte es aquella que puede existir de manera independiente.

Las entidades se representan en el diagrama Entidad-Relación mediante un rectángulo etiquetado en su interior mediante un nombre.

- **Relación**

Una relación es una correspondencia o asociación entre dos o más entidades que se representa en el diagrama Entidad-Relación a través de un rombo etiquetado en su interior con un nombre descriptivo de la función desempeñada por la relación.

En función del número de instancias (ejemplares de una entidad) de una entidad que participan en una relación, éstas pueden ser de tres tipos de cardinalidades: [Card.Relación]

- **Relaciones uno a uno**, donde una instancia de la entidad A se relaciona con una y solamente una de la entidad B. Este tipo de cardinalidad se representa en la relación situando un “1” cerca de la línea que conecta cada entidad con la relación.
- **Relaciones uno a muchos**, donde cada instancia de la entidad A se relaciona con varias instancias de la entidad B. Este tipo de cardinalidad se representa en la relación situando un “1” en la línea que conecta la entidad A con la relación y una “N” en la línea que conecta la entidad B con la relación.

- **Relaciones muchos a muchos**, donde cualquier instancia de la entidad A se relaciona con cualquier instancia de la entidad B. Este tipo de cardinalidad se representa en la relación situando una “N” en la línea que conecta la entidad A con la relación y una “M” en la línea que conecta la entidad B con la relación.

- **Atributo**

Los atributos representan las propiedades básicas o relevantes de una entidad o relación y se representan en un diagrama Entidad-Relación a través de círculos o elipses etiquetados con un nombre en su interior.

Los atributos pueden ser simples o compuestos. Un atributo simple es un atributo que no se puede dividir en partes más pequeñas que tengan un significado propio mientras que un atributo compuesto es aquel que está constituido por varios componentes con significado por sí mismos. Asimismo, los atributos también pueden ser derivados, que son aquellos que representan un valor que se puede obtener a partir del valor de uno o varios atributos.

Una vez realizada una introducción de carácter general sobre el diagrama Entidad-Relación y sus componentes, se va a proceder a presentar, en la figura mostrada en el archivo *Diagrama_E-R.png* del CD adjunto, el diagrama Entidad-Relación correspondiente al modelo de datos de la Asociación Navarra de Síndrome de Down obtenido a partir del estudio de los requisitos funcionales requeridos por el personal de la Asociación y a partir del cual se implementará la Base de Datos que permita almacenar toda la información gestionada por la entidad.

Después de haber presentado en la figura anterior el diagrama Entidad-Relación correspondiente al modelo de datos de la Asociación Navarra de Síndrome de Down se van a explicar a continuación las entidades que lo constituyen así como las relaciones y los principales atributos de cada una de ellas.

- **Personas**

A través de esta entidad se van a almacenar todos los datos de carácter general asociados a todas las personas. En el caso concreto de la Asociación Navarra de Síndrome de Down las personas van a ser solamente los alumnos con Síndrome de Down existentes en la Asociación, así como los profesionales y voluntarios que trabajan o colaboran con la entidad y los socios de la misma. Los atributos de los que constará la entidad “Personas” son los siguientes:

- *Id*: clave que identifica de manera única a cada persona. Se trata de un número entero autoincrementable que se asigna de manera automática a cada persona cuando se da de alta en la Asociación.
- *Nombre_completo*: se trata de un atributo compuesto que representa el nombre de la persona formado por nombre, primer apellido y segundo apellido.
- *DNI*: atributo para almacenar el Documento Nacional de Identidad de la persona en cuestión.
- *SS*: atributo para almacenar el número de la Seguridad Social de la persona.
- *Fecha_nac*: fecha de nacimiento de la persona.
- *Profesión*: profesión de la persona.
- *Tfno1*, *Tfno2* y *Tfno3*: atributos para almacenar los diferentes teléfonos con los que poder contactar a una determinada persona.
- *E-mail1*, *E-mail2*: atributos que almacenarán las direcciones de correo electrónico a las que se podrá enviar información a cada persona.
- *Fax*: número de fax al que se podrá enviar información a cada persona.
- *Comentarios*: atributo de carácter genérico en el que se podrá almacenar cualquier información relevante sobre una persona y que no se corresponda con ninguno de los atributos anteriores.

- **Usuarios**

Dentro de esta entidad se van a considerar todas las personas con Síndrome de Down que acuden a la Asociación. Como se habrá podido observar en el diagrama Entidad-Relación, esta entidad se encuentra relacionada a través de una jerarquía con la entidad “Personas” lo que significa que, debido a que un usuario es una persona, la entidad “Usuarios” va a contar con todos los atributos existentes en la entidad “Personas” además de los que se indican a continuación como atributos propios únicamente de los alumnos con Síndrome de Down.

- *Id_usuario*: clave que identifica de manera única a cada usuario de la Asociación.
- *Nº hermanos*: atributo para almacenar el número de hermanos del alumno con Síndrome de Down.
- *Escuela*: nombre del colegio en el que cursa sus estudios el alumno con Síndrome de Down.
- *Posición*

- *Tutor*: atributo empleado para almacenar el nombre del tutor o persona legal al cargo del alumno, en el caso de que no tuviera padres.
- *DNI_tutor*: atributo para almacenar el Documento Nacional de Identidad del tutor o persona legal al cargo del alumno, en el caso de que no tuviera padres.

- **Socios**

Entidad empleada para representar a aquellas personas que desean participar o colaborar con la Asociación a través del pago de una determinada cuota. Al igual que en el caso de la entidad “Usuarios”, la entidad “Socios” mantiene una relación de jerarquía con la entidad “Personas” puesto que todo socio es una persona y, en consecuencia, contará con todos los atributos propios de una persona. Adicionalmente, la entidad “Socios” cuenta con los siguientes atributos que la identifican de manera única:

- *Id_socio*: clave que identifica de manera única a cada socio de la Asociación.
- *Tipo_socio*: atributo para identificar si el socio es ordinario o colaborador. Un socio ordinario es aquel que se hace cargo de los gastos derivados de las actividades realizadas por éste mientras que un socio colaborador es cualquier otro socio que no hace frente a los gastos de un alumno.
- *Cuota*: atributo para almacenar la aportación realizada por el socio a la Asociación. En el caso de los socios ordinarios esta cuota vendrá fijada por la Asociación mientras que en el caso de los socios colaboradores será una aportación libre.
- *Nº_cta*: número de la cuenta bancaria a la que se remitirán los pagos de las cuotas y/o gastos derivados de las actividades del alumno con Síndrome de Down.
- *Frec_pago*: atributo para indicar la periodicidad con la que el socio realiza el pago de la cuota.
- *Nombre_empresa*: en el caso de que el socio que colabora con la Asociación sea una empresa el nombre de la misma se almacenará en este atributo.
- *CIF*: en el caso de que el socio que colabora con la Asociación sea una empresa el Código de Identificación Fiscal de la misma se almacenará en este atributo.

Al presentar el atributo *Tipo_socio* se ha indicado que los socios ordinarios son aquellos que hacen frente a los gastos de un determinado alumno de la Asociación. Por ello, es necesario que el modelo de datos permita conocer de qué alumno es familiar cada socio y para ello, como se observa en la figura correspondiente al diagrama Entidad-Relación, existe la relación “*Familiar_de*” que indicará de qué usuario es familiar cada socio (ordinario o colaborador). Se trata de una relación “N:M” lo que significa que será posible que un mismo socio pueda ser familiar de más de un alumno y que un mismo alumno puede tener más de un familiar como socio (uno de ellos obligatoriamente será socio ordinario y el resto serán colaboradores).

- **Profesionales**

Entidad que representa a todos aquellos profesores que imparten actividades dentro de la Asociación Navarra de Síndrome de Down. Los atributos que permiten identificar a la entidad, además de los propios de la entidad “Personas” por encontrarse la entidad “Profesionales” relacionada jerárquicamente con ella, son los siguientes:

- *Id_prof*: clave que identifica de manera única a cada profesional de la Asociación.
- *Titulación*: atributo para almacenar la titulación que posee el profesional.
- *Puesto*: atributo para almacenar el cargo que ocupa el profesional dentro de la Asociación.
- *Nº_cta*: número de la cuenta bancaria en la que se ingresará la nómina del profesional o cualquier otra aportación económica que se le realice.
- *Interno/Externo*: los profesionales que trabajan en la Asociación pueden hacerlo porque sean asalariados de la propia Asociación (internos) o bien porque lo hagan a través de acuerdos de colaboración (externos). El atributo “*Interno/Externo*” se emplea para almacenar dicha información.

- **Histórico_gastos_profesionales**

Dentro de esta entidad se consideran todos aquellos gastos asociados a un profesional y derivados de la actividad llevada a cabo por el mismo dentro de la Asociación que deberán remunerarse al profesional. Así, por ejemplo, a través de esta entidad se podrán almacenar los gastos derivados de desplazamientos, alojamientos, comidas, etc. que el profesional deba realizar como consecuencia de su trabajo en la Asociación. Debido a que se trata de una entidad que almacena los gastos asociados a un profesional, nos encontramos ante una entidad débil ya que, como es lógico, si no existen profesionales que generen los gastos, no será necesario disponer de una entidad para almacenarlos. Como atributos identificativos de la entidad se encuentran los siguientes:

- *Concepto*: atributo empleado para indicar a qué se debe el gasto realizado por el profesional.
- *Cantidad*: atributo para almacenar la cantidad de conceptos que deben remunerarse al profesional.
- *Observaciones*: atributo de carácter genérico en el que se podrá almacenar cualquier información relevante sobre el gasto llevado a cabo por un profesional.

Como bien se ha indicado anteriormente, la entidad “Histórico_gastos_profesionales”, almacena los gastos adicionales que deben remunerarse a un determinado profesional. Por ello, es necesaria la existencia de una relación entre esta entidad y la entidad “Profesionales” que indique a qué profesional corresponde cada gasto. Así, en el diagrama Entidad-Relación existe la relación “*Extras_profesional*” con el atributo “*Fecha*” para indicar la fecha en la que se generó el gasto. Como es lógico, cada gasto existente en la entidad solo puede pertenecer a un profesional mientras que un mismo profesional puede haber generado varios gastos por lo que nos encontramos ante una relación “1:N”.

- **Profesionales_precio_hora**

En ocasiones puede ocurrir que alguno de los profesionales contratados por la Asociación para realizar una actividad en concreto tenga capacidades para poder realizar cualquier otra actividad. Por ello, la Asociación se encuentra interesada en conocer el precio de dichos profesionales para impartir actividades extras y es esta información la que se gestiona a través de la entidad “Profesionales_precio_hora” que, debido a que depende de la existencia de la entidad “Profesionales” para existir, se trata de una entidad débil que cuenta con los siguientes atributos:

- *Concepto*: atributo empleado para indicar la actividad extra que podría ser impartida por el profesional.
- *Curso*: atributo para almacenar el curso académico en el que se impartiría la actividad extra.
- *Precio/hora*: precio por hora al que el profesional impartiría la actividad extra.

Al igual que ocurría en el caso de la entidad “Histórico_gastos_profesionales”, en este caso también es necesario conocer qué profesionales impartirían qué actividad extra para lo que existe en el diagrama Entidad-Relación la relación “Cobra” entre las entidades “Profesionales” y “Profesionales_precio_hora”. Se trata de una relación “1:N” debido a que un mismo profesional puede encontrarse capacitado para impartir más de una actividad extra, pero una misma actividad sólo puede impartirse por un profesional.

- **Salarios**

Entidad a través de la que se van a almacenar todos los salarios remunerados a cada profesional cada mes. Se trata de una entidad débil, ya que su existencia se encuentra supeditada a la existencia de la entidad “Profesionales”, que cuenta con los siguientes atributos:

- *Mes*: mes al que corresponde el salario remunerado al profesional.
- *Anho*: año al que corresponde el salario remunerado al profesional.
- *Bruto*: atributo para almacenar el salario bruto que se paga al profesional.
- *Seg_social*: atributo para almacenar el importe que corresponde a la Seguridad Social del trabajador.
- *SS_empresa*: atributo para indicar el importe que debe aportar la empresa a la Seguridad Social.
- *IRPF*: atributo para almacenar el importe correspondiente al IRPF.

De la misma manera que en los dos casos anteriores, es necesaria la existencia de una relación entre las entidades “Profesionales” y “Salarios” que permita conocer a qué profesional pertenece cada uno de los salarios almacenados en la Base de Datos. Debido a que cada salario sólo puede pertenecer a un profesional, pero un profesional recibe un salario cada mes nos encontramos ante una relación “1:N”.

- **Voluntarios**

Se va a considerar como voluntario a cualquier persona que colabore desinteresadamente en la realización de las actividades propuestas desde la Asociación sin ser profesional contratado por la entidad. Al igual que ocurría en el caso de las entidades “Usuarios”, “Socios” y “Profesionales”; la entidad “Voluntarios” depende jerárquicamente de la entidad “Personas” puesto que todo voluntario es una persona. Por este motivo, la entidad “Voluntarios” va a contar con todos los atributos existentes en la entidad “Personas” además de los indicados a continuación como atributos propios de un voluntario:

- *Id_voluntario*: clave que identifica de manera única a cada voluntario de la Asociación.
- *Titulación*: atributo para almacenar la titulación que posee el voluntario.
- *Tipo*: atributo para almacenar el tipo de colaboración que puede realizar el voluntario dentro de la Asociación.
- *Disponibilidad*: atributo de carácter genérico donde se podrá introducir los días y horas en los que el voluntario puede colaborar con la Asociación.
- *Observaciones*: atributo de carácter genérico en el que se podrá almacenar cualquier información relevante sobre un voluntario y que no se corresponda con ninguno de los atributos anteriores.

- **Histórico_gastos_voluntarios**

Al igual que ocurría en el caso de los profesionales podría ocurrir que los voluntarios, para poder llevar a cabo su colaboración con la Asociación, tuvieran que realizar gastos adicionales que correrían a cargo de la entidad como podrían ser gastos de desplazamiento, dietas, etc. Este tipo de gastos son los que se consideran dentro de la entidad “Histórico_gastos_voluntarios” que, debido a que para que exista un gasto derivado de la actividad de un voluntario es necesario que exista dicho voluntario, se trata de una entidad débil que depende de la existencia de la entidad “Voluntario” para existir. Los atributos con los que cuenta esta nueva entidad son los siguientes:

- *Concepto*: atributo empleado para indicar a qué se debe el gasto realizado por el voluntario.
- *Cantidad*: atributo para almacenar la cantidad de conceptos que deben remunerarse al voluntario.
- *Observaciones*: atributo de carácter genérico en el que se podrá almacenar cualquier información relevante sobre el gasto llevado a cabo por un voluntario.

De la existencia de las entidades “Voluntario” e “Histórico_gastos_voluntarios”, como se puede observar en la imagen correspondiente al diagrama Entidad-Relación, se deriva la existencia de una relación que indique a qué voluntario corresponde cada uno de los gastos existentes en la entidad “Histórico_gastos_voluntarios”. Dicha relación recibe el nombre de “*Extras_voluntario*” y se trata de una relación “1:N” (puesto que cada uno de los gastos sólo puede pertenecer a un voluntario mientras que un voluntario puede haber generado varios de los gastos existentes) que cuenta con un atributo “*Fecha*” en el que se almacenará la fecha en la que se genera el gasto correspondiente.

- **Periodos_personas**

Uno de los requisitos funcionales existentes para el desarrollo de la aplicación era que una misma persona podía desempeñar en la Asociación varios roles diferentes al mismo tiempo así como que una misma persona podía darse de alta o de baja varias veces en la Asociación para cada uno de los diferentes roles. Derivado de este requisito, surge la entidad “Periodos_personas” que almacenará los periodos de tiempo en los que cada persona se encuentra activa en cada uno de los roles existentes (usuario, socio, profesional o voluntario). Como se puede observar en la figura correspondiente al diagrama Entidad-Relación, se trata de una entidad débil puesto que si no existen personas no será posible gestionar los periodos en los que éstas se encuentran activas que cuenta con los siguientes atributos:

- *Tipo_persona*: indica el rol (usuario, socio, profesional o voluntario) en el que se da de alta o de baja la persona.
- *Fecha_alta*: atributo para almacenar la fecha en la que una persona se da de alta en un determinado rol.
- *Fecha_baja*: atributo para almacenar la fecha en la que una persona se da de baja en un determinado rol.

De la presentación de la entidad “Periodos_personas” que se acaba de realizar y de la presentación de la entidad “Personas” que se realizó en su momento se deriva la necesidad de la existencia de una relación entre ambas que permita conocer a qué persona corresponde cada uno de los periodos de alta y/o baja existentes en la entidad de periodos. Como cada una de las personas puede darse de alta o de baja varias veces para un mismo rol, pero cada uno de los periodos existentes en la entidad “Periodos_personas” sólo puede pertenecer a una única persona, se tratará de una entidad “1:N” representada en el diagrama Entidad-Relación con el nombre “Activa_por” y formada por el atributo “Observaciones” donde se almacenará cualquier comentario que se considere oportuno y de interés para la gestión de los periodos de activación de cada persona.

- **Programas**

Esta entidad hace referencia a todos y cada uno de los programas de trabajo presentados por la Asociación con objeto de integrar a los alumnos con Síndrome de Down en el mundo deportivo, tecnológico, social, laboral, etc. La entidad “Programas” estará formada por los siguientes atributos identificativos:

- *Id_programa*: clave que identifica de manera única a cada programa ofrecido por la Asociación.
- *Nombre*: nombre del programa en cuestión.
- *Curso*: atributo que almacena el curso académico en el que se oferta el programa.
- *Descripción*: atributo de carácter genérico para almacenar la descripción del programa.
- *Responsable*: atributo empleado para almacenar el nombre de la persona encargada del programa.

- **Actividades**

Dentro de esta entidad se considerarán todas las actividades promovidas por la Asociación dentro de cada programa. Dichas actividades serán identificadas a través de los siguientes atributos:

- *Id_act*: clave que identifica de manera única a cada actividad promovida por la Asociación.
- *Nombre*: nombre dado a la actividad.
- *Comentarios*: atributo empleado para almacenar cualquier observación que se desee realizar sobre la actividad.

Como se acaba de indicar al presentar la entidad “Actividades”, toda actividad promovida por la Asociación se enmarca dentro de alguno de los programas ofertados por la citada Asociación por lo que es necesaria la existencia de una relación entre ambas entidades que permita conocer, para toda actividad, el programa al que pertenece. En el caso que nos ocupa dicha relación es la relación “*pertenece*” que presenta cardinalidad “M:N” debido que una misma actividad puede ofertarse para distintos programas al mismo tiempo que un programa puede estar constituido por varias actividades.

- **Información_actividades**

Cada actividad dispondrá de una serie de información que permitirá conocer en cada momento los datos relativos a la impartición de la citada actividad y que serán aquellos que se gestionen en la entidad “Información_actividades”. Debido a que esta entidad almacena información relativa a una determinada actividad, será necesario que dicha actividad exista para que pueda existir la información vinculada a la misma. Por este motivo, nos encontramos ante una entidad débil que tendrá los siguientes atributos principales:

- *Curso*: atributo que almacena el curso académico en el que se imparte la actividad.
- *Grupo*: es posible que de una misma actividad exista más de un grupo debido al volumen de usuarios apuntados en la misma. En este atributo de la entidad se almacenará el nombre del grupo de la actividad.
- *Lugar*: atributo para almacenar el lugar en el que se llevará a cabo la actividad.
- *Horario*: atributo donde se almacenará los días y horas en los que se impartirá la actividad.
- *Precio*: precio que cuesta realizar la actividad.
- *Responsable*: atributo empleado para almacenar el nombre de la persona encargada de la actividad.

De la existencia de las entidades “Actividades” e “Información_actividades” se deriva la existencia de una relación entre ambas (relación “*Contiene*” del diagrama Entidad-Relación) que permita conocer a qué actividad corresponde cada uno de los registros de información existentes en la entidad “Información_actividades”. Se trata de una relación “1:N” donde cada actividad puede contar con varios registros de información, pero cada uno de estos registro sólo puede encontrarse vinculado a una actividad.

Al observar el diagrama Entidad-Relación se comprueba que tanto la entidad “Usuarios” como las entidades “Profesionales” y “Voluntarios” se encuentran relacionadas con la entidad “Información_actividades” y es que es necesario conocer qué alumnos con Síndrome de Down realizan qué actividades y qué profesionales y/o voluntarios imparten o colaboran en qué actividades. Debido a que es posible que un mismo alumno, profesional o voluntario realice, imparta o colabore en más de una actividad y que una actividad sea cursada o impartida por más de un alumno, profesional o voluntario las tres relaciones existentes son “N:M” que permiten conocer a través de sus atributos los periodos o fechas concretas en las que un alumno, profesional o voluntario se ha encontrado relacionado con una determinada actividad.

- **Periodos_actividades**

A través de esta nueva entidad se van a considerar todos aquellos periodos en los que se encuentra activo cada uno de los diferentes grupos existentes para cada actividad y es que puede ser posible que un grupo de una actividad se dé de alta o de baja varias veces a lo largo de su existencia. Los atributos con los que cuenta la entidad “Periodos_actividades” para poder gestionar los periodos de existencia de cada grupo de cada actividad son los siguientes:

- *Fecha_alta*: atributo para almacenar la fecha en la que un grupo de una actividad se da de alta.
- *Fecha_baja*: atributo para almacenar la fecha en la que un grupo de una actividad se da de baja.

Como ya se habrá podido deducir, la entidad “Periodos_actividades” es una entidad débil puesto que su existencia se encuentra supeditada a la existencia de la entidad “Información_actividades” de manera que existan grupos a los que poder dar de alta o de baja.

Vinculada a las entidades “Información_actividades” y “Periodos_actividades” se encuentra la relación “*Impartida_en*” que permite conocer en qué periodos de tiempo se ha impartido un determinado grupo de una actividad. Se trata de una relación “1:N” que permite que un mismo grupo de una actividad se encuentre relacionado con varios periodos puesto que puede darse de alta o de baja varias veces, pero un periodo sólo se encuentre relacionado con un grupo de una actividad.

- **Proveedores**

Esta entidad hace referencia a todos aquellos proveedores que suministran cualquier tipo de material a la Asociación. Los atributos que identifican a las instancias de la entidad “Proveedores” son los siguientes:

- *Id_prov*: clave que identifica de manera única a cada proveedor.
- *Nombre*: atributo que almacena el nombre del proveedor.
- *CIF*: atributo que almacena el Código de Identificación Fiscal del proveedor.
- *Persona_contacto*: nombre del trabajador de la empresa proveedora con el que contacta la Asociación.
- *Tfno_contacto*: teléfono a través del que se contacta con la empresa proveedora.
- *Email_contacto*: correo electrónico a través del que se contacta con la empresa proveedora.

- *Web*: atributo que almacena la URL de la página Web de la empresa proveedora.
- *Observaciones*: atributo de carácter genérico donde se podrá introducir cualquier información del proveedor que se considere de interés.

- **Facturas**

Entidad que representa las facturas que han sido remitidas a la Asociación Navarra de Síndrome de Down por alguno de sus proveedores. Las instancias de esta entidad se identifican a través de los siguientes atributos:

- *Id_fact*: clave única que identifica a la factura dentro del sistema de gestión de la propia Asociación.
- *Id_fact_prov*: clave única que identifica a la factura dentro del sistema de gestión del proveedor remitente de la factura.
- *Fecha*: atributo que almacena la fecha de expedición de la factura.
- *Concepto*: atributo que almacena el concepto por el que se expide la factura.
- *Importe*: atributo que almacena el importe al que asciende la factura.
- *Estado*: atributo que almacena el estado en el que se encuentra la factura (grabada, enviada a la asesoría, contabilizada, etc.).

Todas y cada una de las facturas gestionadas por la Asociación han tenido que ser remitidas por alguno de los proveedores con los que trabaja la entidad por lo que es importante conocer qué facturas fueron generadas por cada proveedor. Para ello existe en el diagrama Entidad-Relación la relación “*expide*” que permite conocer dicha información y cuenta con cardinalidad “1:N” debido a que un proveedor ha podido generar más de una factura, pero una misma factura sólo ha podido ser generada por un proveedor.

- **Lineas_factura**

La entidad “Lineas_factura” hace referencia a todas y cada una de las líneas que componen una factura. Se trata, por tanto, de una entidad débil ya que si no existe la factura correspondiente no va a ser posible que existan líneas asociadas a ella. Los atributos de los que consta la entidad son los siguientes:

- *Id_lin_fact*: clave que identifica de manera única cada línea dentro de una factura.
- *Descripción*: atributo que almacena el concepto por el que se genera la línea de factura.
- *Cantidad*: atributo que almacena el número de materiales comprados en cada línea de factura.

Con objeto de conocer a qué factura pertenece cada una de las líneas instanciadas en la entidad “Lineas_factura” existe la relación “*tiene*” entre dicha entidad y la entidad “Facturas”. Dicha relación es una relación “1:N” puesto que una línea de factura sólo puede pertenecer a una factura mientras que una misma factura puede estar compuesta por varias líneas.

- **Instituciones**

A través de esta entidad se van a almacenar todos los datos de carácter general de aquellas entidades públicas o privadas que colaboran con la Asociación. Los atributos de los que constará la entidad “Instituciones” son los siguientes:

- *Id*: clave que identifica de manera única a cada entidad.
- *Nombre*: atributo que almacena el nombre de la entidad que colabora con la Asociación.
- *Departamento*: atributo que almacena el nombre del departamento al que pertenece la entidad colaboradora, en el caso de que se trate de una entidad pública.
- *Tfno*: número de teléfono a través del cual se podrá contactar con la entidad colaboradora.
- *Email*: correo electrónico a través del cual se podrá contactar con la entidad colaboradora.
- *Web*: atributo que almacena la URL correspondiente a la página Web de la entidad colaboradora.

- **Subvenciones**

Entidad que representa todas y cada una de las subvenciones que han sido concedidas a la Asociación Navarra de Síndrome de Down por parte de alguna de las entidades existentes en la entidad “Instituciones”. Los atributos existentes para la entidad son los siguientes:

- *Nombre*: nombre de la subvención concedida a la Asociación.
- *Anho*: atributo que almacena el año en el que se concede la subvención.
- *Tipo*: atributo que almacena el tipo de subvención.
- *Estado*: atributo que almacena el estado en el que se encuentra la subvención (tramitada, concedida, etc.).
- *Presupuestado*: cantidad que la Asociación presupuesta como necesaria para la subvención.
- *Solicitado*: cantidad que la Asociación solicita para la subvención.
- *Concedido*: cantidad que se concede en la subvención.
- *Fecha_solicitud*: atributo que almacena la fecha en la que se concede la subvención.

Como ya se ha comentado al realizar la descripción de la entidad, las subvenciones son concedidas a la Asociación por parte de alguna de las instituciones que colaboran con la misma. Por lo tanto, será necesario poder conocer qué subvenciones han sido concedidas por qué instituciones y para ello existe la relación “Concede” que se trata de una relación 1:N entre las entidades “Instituciones” y “Subvenciones” ya que una institución puede conceder varias subvenciones, pero una subvención únicamente es concedida por una institución.

Asimismo, y también relacionado con la entidad “Subvenciones”, el personal administrativo de la Asociación necesita conocer qué facturas y qué salarios se presentan a cada subvención. Con este motivo existen las relaciones “*destinan*” y “*presentada*” entre las entidades “Subvenciones” y “Profesionales” en el primer caso y entre las entidades “Subvenciones” y “Facturas” en el segundo. Ambas relaciones son N:M puesto que, al poder presentar porcentajes parciales tanto de los salarios como de las facturas a cada subvención, es posible que una misma factura o salario se presente a más de una subvención y, como es lógico, a una misma subvención se pueden presentar varios salarios o facturas. Los atributos de los que constan ambas relaciones son los siguientes:

- *Porcentaje*: atributo que almacena el porcentaje del total de una factura o del total del salario anual de un profesional que se presenta a la subvención.
- *Importe*: atributo calculado en función del valor del atributo “*Porcentaje*” y del salario total de un profesional o del importe total de una factura. El valor almacenado en este atributo indicará el valor de la factura o del salario que se aporta a la subvención.
- *Fecha*: atributo que almacena la fecha en la que el salario o factura se presenta a la subvención.

- **Direcciones**

Dentro de esta entidad se van a considerar todas las direcciones físicas en las que tienen fijado su domicilio las personas (usuarios, socios, profesionales y voluntarios) relacionadas con la Asociación o en las que se encuentran ubicadas las entidades que colaboran con la citada Asociación. Los atributos a través de los cuales va a ser posible definir una dirección son los siguientes:

- *Id*: clave que identifica de manera única a cada dirección.
- *Calle*: atributo que almacena el nombre de la calle en la que reside una persona o se encuentra ubicada una entidad.
- *Nº*: atributo que almacena el número dentro de la calle en el que reside una persona o se encuentra ubicada una dirección.
- *Escalera*: atributo que identifica la escalera, dentro de un mismo piso, en la que reside una persona o se encuentra ubicada una entidad.
- *Piso*: atributo que almacena la altura del piso en la que reside una persona o se encuentra ubicada una entidad.
- *Letra*: atributo que identifica, dentro de un mismo piso, la letra en la que reside una persona o se encuentra ubicada una entidad.
- *CP*: código postal de la población a la que corresponde la dirección.
- *Población*: atributo que almacena la localidad en la que reside una persona o se encuentra ubicada una entidad.
- *Provincia*: atributo que almacena la provincia en la que reside una persona o se encuentra ubicada una entidad.

Para poder conocer en qué dirección de todas las existentes en la Asociación reside cada persona o se encuentra ubicada cada institución colaboradora es necesario que exista una relación entre las entidades “Persona” o “Institución” y la entidad “Direcciones”. Dichas relaciones reciben el nombre de “Vive_en” en el primer caso y “Ubicada” en el segundo y son relaciones 1:N puesto que una misma persona o institución solamente puede estar relacionada con una dirección, pero en una misma dirección es posible que residan varias de las personas o instituciones vinculadas a la Asociación.

- **Dominios**

A lo largo de la presente memoria se ha indicado en varias ocasiones que algunos de los atributos existentes en las entidades que representan el modelo de la Asociación sólo podrán tomar valores que se encuentren previamente definidos. Atendiendo a esta especificación, se ha considerado oportuno agrupar dichos valores en diferentes categorías de manera que al seleccionar un valor predefinido para un determinado atributo solo se muestren como posibles opciones aquellos que pertenezcan a la categoría correspondiente. Así, la entidad “Dominios” que en este momento nos ocupa hace referencia a todas aquellas categorías definidas para agrupar los posibles valores que pueda tomar un determinado atributo. La definición de los citados dominios se realizará a través de los dos atributos siguientes:

- *Id*: clave que identifica de manera única a cada dominio o categoría de valores predefinidos.
- *Descripción*: atributo que almacena el nombre del dominio.

- **Descripciones**

Siguiendo con el razonamiento expuesto al presentar la entidad “Dominios”, podemos decir que la entidad “Descripciones” hace referencia a todos aquellos valores predefinidos que podrán seleccionarse como valor para un determinado atributo. La definición de cada descripción vendrá dada por los siguientes campos:

- *Id*: clave que identifica de manera única a cada valor predefinido.
- *Descripción*: atributo que almacena el contenido del valor predefinido.

Con la intención de poder conocer a qué dominio pertenece cada descripción existe la relación “*Contiene*” entre las entidades “Dominios” y “Descripciones” de manera que sea posible conocer a qué dominio pertenece cada descripción y así saber qué valores, de entre todos los predefinidos, se deben mostrar a la hora de dar valor a cada atributo. Se trata de una relación 1:N puesto que una descripción solo puede pertenecer a un dominio o categoría, pero una categoría puede estar formada por varias descripciones.

- **Tarifas_gastos**

Esta entidad hace referencia a las tarifas establecidas que se deberán pagar a un empleado por los gastos adicionales derivados de su trabajo. Los atributos con los que va a contar esta entidad son los siguientes:

- *Id*: clave única que va a permitir identificar cada tarifa.
- *Concepto*: descripción del tipo de gasto a recompensar económicamente.
- *Anho*: año en el que se aplica el valor indicado en el atributo “*Precio*” como tarifa a recompensar por un gasto.
- *Precio*: precio que deberá recompensarse por el gasto.

- **Cuotas**

Entidad que representa las cuotas que deberá pagar cada año cada uno de los socios ordinarios de la Asociación Navarra de Síndrome de Down. La entidad “Cuota” dispone de los siguientes atributos:

- *Anho*: año en el que se aplica el valor indicado en el atributo “*Importe*” como cuota a pagar por los socios ordinarios de la Asociación.
- *Concepto*: descripción de la cuota de cada año.
- *Importe*: precio que deberá abonar cada socio ordinario por la cuota del año correspondiente.

- **Índices**

Entidad que va a permitir que las claves que identifican a usuarios, socios, profesionales y voluntarios sean únicas y se incrementen de uno en uno. Los atributos necesarios para conseguir dicho propósito son los siguientes:

- *Ind_socio*: atributo que almacena el identificador dado al último socio creado en la herramienta.
- *Ind_usuario*: atributo que almacena el identificador dado al último usuario creado en la herramienta.
- *Ind_voluntario*: atributo que almacena el identificador dado al último voluntario creado en la herramienta.
- *Ind_profesional*: atributo que almacena el identificador dado al último profesional creado en la herramienta.

- **Personal**

Se va a considerar como personal a cualquier trabajador administrativo de la Asociación Navarra de Síndrome de Down con un nombre y una contraseña válidos para acceder al *software*. Esta entidad tendrá los siguientes atributos:

- *Id_personal*: clave única que identifica al personal administrativo de la Asociación.
- *Nombre*: nombre del administrativo de la Asociación que accede a la herramienta.
- *Login*: nombre con el que el administrativo de la Asociación accede a la herramienta.
- *Password*: clave de acceso con la que el administrativo de la Asociación accede a la herramienta.

- **Perfiles**

Entidad que representa los diferentes perfiles de acceso con los que el personal administrativo de la Asociación podrá acceder a la herramienta. Los atributos necesarios para poder gestionar dicho acceso son los siguientes:

- *Id_perfil*: clave única que identifica a un determinado perfil.
- *Nombre*: nombre dado al perfil de acceso a la herramienta.
- *Menu_personas*: atributo que indica el tipo de acceso (sin acceso, lectura o escritura) que el perfil tiene sobre la opción “Personas” del menú.
- *Menu_programas*: atributo que indica el tipo de acceso (sin acceso, lectura o escritura) que el perfil tiene sobre la opción “Programas” del menú.
- *Menu_actividades*: atributo que indica el tipo de acceso (sin acceso, lectura o escritura) que el perfil tiene sobre la opción “Actividades” del menú.
- *Menu_usuarios_acts*: atributo que indica el tipo de acceso (sin acceso, lectura o escritura) que el perfil tiene sobre la opción “Usuarios-Actividades” del menú.
- *Menu_instituciones*: atributo que indica el tipo de acceso (sin acceso, lectura o escritura) que el perfil tiene sobre la opción “Entidades” del menú.
- *Menu_subvenciones*: atributo que indica el tipo de acceso (sin acceso, lectura o escritura) que el perfil tiene sobre la opción “Subvenciones” del menú.
- *Menu_proveedores*: atributo que indica el tipo de acceso (sin acceso, lectura o escritura) que el perfil tiene sobre la opción “Proveedores” del menú.
- *Menu_facturas*: atributo que indica el tipo de acceso (sin acceso, lectura o escritura) que el perfil tiene sobre la opción “Facturas” del menú.
- *Menu_salarios*: atributo que indica el tipo de acceso (sin acceso, lectura o escritura) que el perfil tiene sobre la opción “Salarios” del menú.
- *Menu_gastos_extras*: atributo que indica el tipo de acceso (sin acceso, lectura o escritura) que el perfil tiene sobre la opción “Gastos Extras” del menú.
- *Menu_acts_extras*: atributo que indica el tipo de acceso (sin acceso, lectura o escritura) que el perfil tiene sobre la opción “Actividades Extras” del menú.
- *Menu_perfiles*: atributo que indica el tipo de acceso (sin acceso, lectura o escritura) que el perfil tiene sobre la opción “Perfiles” del menú.
- *Menu_personal*: atributo que indica el tipo de acceso (sin acceso, lectura o escritura) que el perfil tiene sobre la opción “Personal” del menú.
- *Menu_password*: atributo que indica el tipo de acceso (sin acceso, lectura o escritura) que el perfil tiene sobre la opción “Cambio Password” del menú.

Para poder controlar las acciones que cada administrativo de la Asociación puede realizar en la herramienta es necesario que cada uno de los usuarios de acceso a la herramienta se encuentre asociado a uno de los perfiles definidos en la entidad “Perfiles”. Para ello, existe la relación “*tiene_perfil*” entre las entidades “Personal” y “Perfiles” que permite conocer el perfil asignado a cada usuario administrativo de la herramienta. Dicha relación es una relación “1:N” puesto que un usuario de la herramienta sólo puede estar asociado a un perfil, pero un mismo perfil puede tener asociados varios usuarios.

Capítulo 3. Diseño

El presente capítulo de la memoria del proyecto de creación de un sistema informático de gestión Web para la Asociación Navarra de Síndrome de Down se va a dedicar a la presentación del diseño de los diferentes elementos que harán posible que los requisitos (tanto software como hardware y de apariencia), diagramas de casos de uso y diagrama Entidad-Relación realizados en el capítulo de análisis se transformen en un modelo físico capaz de dar cabida y satisfacer todos y cada uno de los aspectos identificados y definidos en el citado capítulo de análisis. De esta manera, en la redacción del capítulo que ahora nos ocupa se va a llevar a cabo el diseño tanto de la arquitectura del sistema del que será necesario disponer para el correcto funcionamiento de la aplicación desarrollada, como el diseño relacional de la Base de Datos al que da lugar el diagrama Entidad-Relación presentado en el capítulo anterior y que será necesario para el correcto almacenamiento y explotación de la información manejada por la Asociación. Finalizaremos el apartado con la presentación del diseño de la interfaz gráfica que se empleará como pauta en la elaboración de todas y cada una de las pantallas con las que cuente la aplicación con el fin de cumplir los requisitos de apariencia establecidos en el capítulo anterior.

Comenzamos la labor de diseño de la aplicación por la presentación de la arquitectura del sistema que permitirá identificar y conocer los diferentes componentes que son necesarios para poder disponer de la aplicación implementada funcionando de manera correcta.

3.1.- Arquitectura del sistema

El diseño de la arquitectura del sistema permite obtener un esqueleto estructurado y jerárquico de las entidades o componentes necesarios para el correcto funcionamiento de la aplicación [Def.Arq.Sistema]. Es decir, representa la estructura de los datos y los componentes que se requieren para poder construir y explotar una aplicación informática. A través de la arquitectura del sistema se podrá conocer, además de los componentes involucrados en el sistema propiamente dicho, las comunicaciones e interrelaciones entre ellos así como los flujos de navegación que se seguirán para el funcionamiento de la aplicación [Def.Arq.Comp] [Def.Arq.Sw]. De acuerdo a la

definición realizada, el diseño de la arquitectura del sistema es un proceso importante en la implementación de una nueva aplicación informática ya que, al proporcionar los detalles completos del software a construir, asegura que la construcción de dicho software sea correcta y completa.

Después de haber realizado una breve definición sobre el diseño de la arquitectura del sistema se va a presentar a continuación la arquitectura propia del sistema de información cuya implementación es objeto del proyecto que nos ocupa.

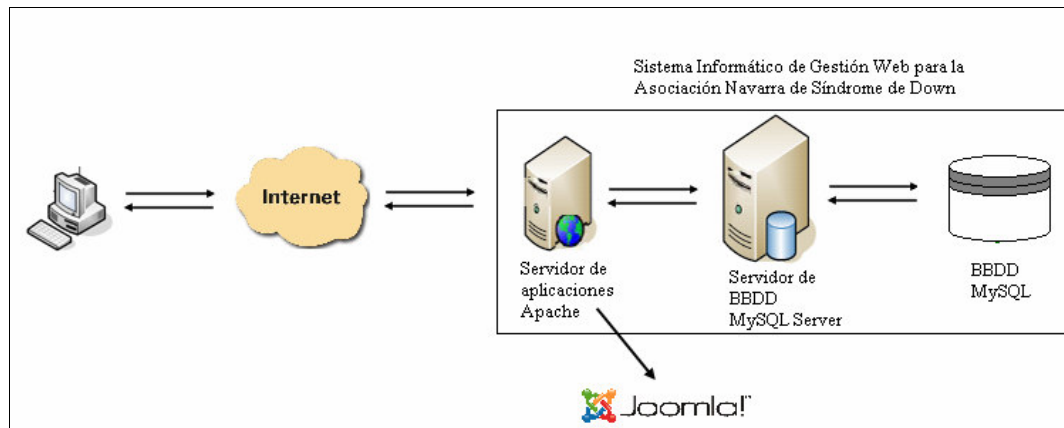


Figura 38. Arquitectura del sistema

Como se observa en la Figura 38, los componentes básicos que constituyen la arquitectura del sistema informático de gestión Web para la Asociación Navarra de Síndrome de Down son los siguientes:

- **Servidor de aplicaciones Apache**

Apache es un servidor Web de distribución libre y de código abierto empleado para servir páginas Web estáticas y dinámicas en la World Wide Web [ApacheDef]. Algunas de las características que hacen de Apache el servidor Web libre de mayor reconocimiento son las siguientes: [ApacheCarac]

- Corre en multitud de Sistemas Operativos.
- Es un servidor altamente configurable de diseño modular por lo que es muy sencillo ampliar sus capacidades.
- Soporta lenguajes de script como pueden ser PHP o Perl.
- Permite personalizar la respuesta ante los posibles errores que se puedan dar en el servidor.

La versión del servidor Apache empleada para alojar los servicios implementados en el proyecto es la 2.0.63 para Windows obtenida de la página oficial de Apache [Apache] e instalada según la configuración por defecto [InstalarApache] [InstalarApacheII].

- **Servidor de BBDD MySQLServer**

Servidor en el que se encuentra alojada la base de datos que almacena toda la información gestionada por el sistema informático implementado. Para la elaboración del proyecto que nos ocupa se ha instado la versión 5.1.41 del servidor MySQLServer obtenida de la página referenciada en [MySQL]. Debido a las carencias existentes para la administración del servidor de bases de datos MySQL se hace necesaria la instalación del software MySQL Administrator en su versión 5.0 [MySQLAdmin] a través del que podremos arrancar y parar el servidor, monitorizarlo y configurarlo. Tanto el servidor como el software de administración fueron instalados siguiendo las indicaciones obtenidas del manual de instalación y configuración encontrado en la referencia [MySQLInst].

- **BBDD MySQL**

La BBDD MySQL con la que cuenta el sistema se emplea para almacenar toda la información manejada por la aplicación implementada. Con objeto de facilitar el manejo de la información en la BBDD se ha instalado el software MySQL Query Browser en su versión 5.0 [MySQLAdmin] que es una herramienta gráfica para crear, ejecutar y optimizar consultas de manera gráfica.

- **Joomla**

Joomla es un gestor de contenidos Web (CMS) que permite crear sitios Web dinámicos e interactivos de manera fácil y sencilla. Se trata de una aplicación gratuita de código abierto programada en lenguaje PHP que utiliza una base de datos MySQL para almacenar el contenido y los parámetros de configuración [EduJoomla]. En el caso que nos ocupa se ha instalado la versión 1.0.15 del CMS obtenida de la página oficial de Joomla en español [JoomlaES] e instalada según el manual de instalación obtenido de la página referenciada en [JoomlaInst].

Después de haber realizado una breve descripción de todos y cada uno de los componentes que constituyen la arquitectura del sistema informático de gestión Web para la Asociación Navarra de Síndrome de Down, se va a describir a continuación cómo interactúa el usuario con el sistema para poder hacer uso de la aplicación así como cómo se interrelacionan los componentes del sistema entre ellos para poder ofrecer la funcionalidad solicitada por el usuario.

Como se puede deducir al observar la Figura 38, el usuario accederá a la aplicación Web a través de Internet. Todas y cada una de las peticiones realizadas por el usuario a través de la aplicación serán atendidas por el servidor de aplicaciones Apache que se comunicará con la base de datos MySQL a través del servidor de bases de datos para poder dar respuesta a las peticiones recibidas. Una vez obtenida la respuesta, ésta se mostrará en la aplicación Web que visualiza el usuario a través de Internet.

3.2.- Diseño relacional de la Base de Datos

Como ya se ha comentado en la introducción de este tercer capítulo, otro de los aspectos a tratar en el diseño de la aplicación a implementar es el correspondiente al diseño relacional de la base de datos que permitirá gestionar toda la información manejada por la Asociación. Dicho diseño relacional se obtiene a partir del diagrama Entidad-Relación generado en el capítulo de análisis en función de los requisitos que debe satisfacer la nueva herramienta por lo que se garantiza que la base de datos a implementar a partir del diseño que se presenta en este capítulo será capaz de gestionar correctamente la información necesaria para el funcionamiento de la Asociación.

Para comenzar con la realización del diseño relacional de la base de datos, y antes de presentar de manera detallada dicho diseño, se va a realizar una breve introducción teórica sobre el proceso que debe seguirse para obtener el modelo relacional de una base de datos a partir del diagrama Entidad-Relación de la misma.

Para convertir un diagrama Entidad-Relación a un modelo relacional basta con partir del diagrama Entidad-Relación y aplicar las normas que se indican a continuación [FichBBDD]:

- Todas las entidades fuertes existentes en el diagrama Entidad-Relación se convierten en tablas en el modelo relacional cuyos campos serán todos y cada uno de los atributos de la entidad correspondiente. Así mismo, los campos que constituirán la clave primaria en la tabla del modelo relacional serán los mismos que eran clave primaria en la entidad del diagrama Entidad-Relación.
- Todas las entidades débiles existentes en el diagrama Entidad-Relación se convierten en tablas en el modelo relacional. Los campos de la tabla serán todos los atributos de la entidad débil en el diagrama Entidad-Relación junto con los atributos que constituyen la clave primaria de la entidad fuerte del diagrama Entidad-Relación de la que depende la entidad débil en cuestión. La clave primaria de la nueva tabla generada estará constituida por los atributos clave de la entidad débil en el diagrama E-R más los atributos clave de la entidad fuerte de la que depende en el diagrama E-R.
- En el caso de que una entidad, ya sea fuerte o débil, en el diagrama E-R contenga atributos compuestos el paso a tabla en el modelo relacional deberá realizarse creando en la nueva tabla tantos campos independientes como atributos contenga el atributo compuesto.
- Las relaciones de cardinalidad 1:N en el diagrama E-R se convierten a modelo relacional mediante propagación de clave. Así, los atributos que constituyen la clave primaria en la entidad de la relación que tenga cardinalidad 1 se convierten en campos de la tabla del modelo relacional generada para la entidad que tiene cardinalidad N (estos campos serán clave foránea de la tabla correspondiente a la entidad de cardinalidad N). En el caso de que la relación 1:N del diagrama E-R contara con atributos propios, estos atributos también se incluirán como campos de la tabla del modelo relacional generado para la entidad con cardinalidad N en la relación.

- Para cada relación de cardinalidad N:M en el diagrama E-R se crea una nueva tabla en el modelo relacional cuyos campos serán los atributos que eran clave primaria en las entidades participantes en la relación además de los atributos propios de la relación en el caso de que los hubiera. Todos los campos de la nueva tabla generada en el modelo relacional serán, al mismo tiempo, clave primaria y foránea en la nueva tabla.
- Las jerarquías existentes en el diagrama E-R se convierten a modelo relacional generando una tabla para cada entidad involucrada en la jerarquía siguiendo los criterios definidos en los puntos anteriores. Además, en el caso de las entidades “hijas” las tablas a las que dan lugar contarán como campo con el atributo que es clave primaria en la entidad “padre” de la que dependen. Dicho campo será clave foránea de la nueva tabla y formará parte de la clave primaria de la misma.

Aplicando los criterios a seguir para convertir un diagrama Entidad-Relación a modelo relacional al diagrama E-R obtenido del análisis de la aplicación a implementar se obtiene el modelo relacional que se muestra en el archivo *Modelo_relacional.png* del CD adjunto como diseño de la base de datos que deberá implementarse para la correcta gestión de la información de la Asociación.

Para finalizar con el diseño correspondiente a la base de datos de la aplicación se van a presentar a continuación, de manera detallada, todas y cada una de las tablas de dicha base de datos obtenidas de manera automática gracias a la herramienta SQL Manager for MySQL [MySQLManager].

- **Tabla Personas**

Table: personas						
Fields						
Name	Type	Not Null	Unique	P/K	A/I	Binary
ID_Persona	int(11)	Not Null		Yes	Yes	
DIRECCIONES_ID_Direccion	int(11)	Not Null				
Nombre	varchar(100)	Not Null				
Primer_Apellido	varchar(100)	Not Null				
Segundo_Apellido	varchar(100)	Not Null				
DNI	varchar(100)	Not Null	Yes			
Num_Seg_Social	varchar(100)					
Fecha_Nacimiento	date					
Telefono1	varchar(100)	Not Null				
Telefono2	varchar(100)					
Telefono3	varchar(100)					
E-mail1	varchar(100)					
E-mail2	varchar(100)					
Fax	varchar(100)					
Profesion	varchar(100)					
Comentarios	longtext					

Figura 40. Tabla Personas

Indices				
Index Name	On Field	Unique	Full Text	Sorting
PRIMARY	`ID_Persona`	Yes		
DNI_UNICO	`DNI`	Yes		
fk_PERSONAS_DIRECCIONES	`DIRECCIONES_ID_Direccion`			
NOMBRE_PERSONA_INDEX	`Nombre`, `Primer_Apellido`, `Segundo_Apellido`			
DNI_INDEX	`DNI`			

Figura 41. Índices tabla Personas

- **Tabla Periodos Personas**

Table: periodos_personas						
Fields						
Name	Type	Not Null	Unique	P/K	A/I	Binary
PERSONAS_ID_Persona	int(11)	Not Null		Yes		
Tipo_Persona	varchar(100)	Not Null		Yes		
Fecha_Alta	datetime	Not Null		Yes		
Fecha_Baja	datetime	Not Null		Yes		
Observaciones	longtext					

Figura 42. Tabla Periodos_personas

Indices				
Index Name	On Field	Unique	Full	Sorting
PRIMARY	`Tipo_Persona`, `Fecha_Alta`, `Fecha_Baja`, `PERSONAS_ID_Persona`	Yes		
fk_PERIODOS_PERSONAS_PERSONAS	`PERSONAS_ID_Persona`			
PERSONA_TIPO_INDEX	`PERSONAS_ID_Persona`, `Tipo_Persona`			

Figura 43. Índices tabla Periodos_personas

- **Tabla Socios**

Table: socios						
Fields						
Name	Type	Not Null	Unique	P/K	A/I	Binary
PERSONAS_ID_Persona	int(11)	Not Null		Yes		
ID_Socio	int(11)	Not Null	Yes			
Tipo_Socio	varchar(100)	Not Null				
Frecuencia_Pago	varchar(100)					
Cuota	float	Not Null				
Numero_Cuenta	varchar(100)	Not Null				
Nombre_Empresa	varchar(100)					
CIF	varchar(100)		Yes			

Figura 44. Tabla Socios

Indices				
Index Name	On Field	Unique	Full Text	Sorting
PRIMARY	`PERSONAS_ID_Persona`	Yes		
SOCIO_UNICO	`ID_Socio`	Yes		
CIF_UNICO	`CIF`	Yes		
fk_SOCIOS_PERSONAS	`PERSONAS_ID_Persona`			
SOCIO_INDEX	`ID_Socio`			
PERSONA_SOCIO_INDEX	`PERSONAS_ID_Persona`			

Figura 45. Índices tabla Socios

- **Tabla Usuarios**

Table: usuarios						
Fields						
Name	Type	Not Null	Unique	P/K	A/I	Binary
PERSONAS_ID_Persona	int(11)	Not Null		Yes		
ID_Usuario	int(11)	Not Null	Yes			
Numero_hermanos	int(11)					
Tutor	varchar(100)					
DNI_Tutor	varchar(100)					
Escuela	varchar(100)					
Posicion	varchar(100)					

Figura 46. Tabla Usuarios

Indices				
Index Name	On Field	Unique	Full Text	Sorting
PRIMARY	`PERSONAS_ID_Persona`	Yes		
USUARIO_UNICO	`ID_Usuario`	Yes		
fk_USUARIOS_PERSONAS	`PERSONAS_ID_Persona`			
USUARIO_INDEX	`ID_Usuario`			
PERSONA_USUARIO_INDEX	`PERSONAS_ID_Persona`			

Figura 47. Índices tabla Usuarios

- **Tabla Socio familiar usuario**

Table: socio_familiar_usuario						
Fields						
Name	Type	Not Null	Unique	P/K	A/I	Binary
SOCIOS_PERSONAS_ID_Persona	int(11)	Not Null		Yes		
USUARIOS_PERSONAS_ID_Persona	int(11)	Not Null		Yes		
Observaciones	longtext					

Figura 48. Tabla Socio_familiar_usuario

Indices					
Index Name	On Field	Unique	Full Text	Sorting	
PRIMARY	`SOCIOS_PERSONAS_ID_Persona`, `USUARIOS_PERSONAS_ID_Persona`	Yes			
fk_SOCIOS_has_USUARIOS_SOCIOS	`SOCIOS_PERSONAS_ID_Persona`				
fk_SOCIOS_has_USUARIOS_USUARIOS	`USUARIOS_PERSONAS_ID_Persona`				
SOCIO_FAMILIAR_INDEX	`SOCIOS_PERSONAS_ID_Persona`				
USUARIO_FAMILIAR_INDEX	`USUARIOS_PERSONAS_ID_Persona`				

Figura 49. Índices tabla Socio_familiar_usuario

- **Tabla Profesionales**

Table: profesionales						
Fields						
Name	Type	Not Null	Unique	P/K	A/I	Binary
PERSONAS_ID_Persona	int(11)	Not Null		Yes		
ID_Profesional	int(11)	Not Null	Yes			
Interno/Externo	varchar(100)	Not Null				
Titulacion	varchar(100)	Not Null				
Puesto	varchar(100)					
Numero_cuenta	varchar(100)	Not Null				

Figura 50. Tabla Profesionales

Indices					
Index Name	On Field	Unique	Full Text	Sorting	
PRIMARY	`PERSONAS_ID_Persona`	Yes			
PROFESIONAL_UNICO	`ID_Profesional`	Yes			
fk_PROFESIONALES_PERSONAS	`PERSONAS_ID_Persona`				
PROFESIONAL_INDEX	`ID_Profesional`				
PERSONA_PROF_INDEX	`PERSONAS_ID_Persona`				

Figura 51. Índices tabla Profesionales

- **Tabla Profesionales precio hora**

Table: profesionales_precio_hora						
Fields						
Name	Type	Not Null	Unique	P/K	A/I	Binary
PROFESIONALES_PERSONAS_ID Persona	int(11)	Not Null		Yes		
Concepto	varchar(100)	Not Null		Yes		
Curso	varchar(100)	Not Null		Yes		
Precio_Hora	float	Not Null				
Observaciones	longtext					

Figura 52. Tabla Profesionales_precio_hora

Indices				
Index Name	On Field	Unique	Full Text	Sorting
PRIMARY	`PROFESIONALES_PERSONAS_ID_ Persona`, `Concepto`, `Curso`	Yes		
fk_PROFESIONALES_PRE CIO_HORA_PROFESIONA LES	`PROFESIONALES_PERSONAS_ID_ Persona`			
PROF_CONCEPTO_CURS O_INDEX	`PROFESIONALES_PERSONAS_ID_ Persona`, `Concepto`, `Curso`			
CONCEPTO_CURSO_IND EX	`Concepto`, `Curso`			

Figura 53. Índices tabla Profesionales_precio_hora

- **Tabla Histórico gastos profesionales**

Table: historico_gastos_profesionales						
Fields						
Name	Type	Not Null	Unique	P/K	A/I	Binary
PROFESIONALES_PERSONAS_ID_Persona	int(11)	Not Null		Yes		
Concepto	varchar(100)	Not Null		Yes		
Fecha	date	Not Null		Yes		
Cantidad	float	Not Null				
Observaciones	longtext					

Figura 54. Tabla Histórico_gastos_profesionales

Indices				
Index Name	On Field	Unique	Full Text	Sorting
PRIMARY	`PROFESIONALES_PERSONAS_ID_Persona`, `Concepto`, `Fecha`	Yes		
fk_HISTORICO_GASTOS_PROFESIONALES	`PROFESIONALES_PERSONAS_ID_Persona`			
PERSONA_GASTOS_PROF_INDEX	`PROFESIONALES_PERSONAS_ID_Persona`			
CONCEPTO_PROF_INDEX	`Concepto`			
PROF_CONCEPTO_INDEX	`Concepto`, `PROFESIONALES_PERSONAS_ID_Persona`			

Figura 55. Índices tabla Histórico_gastos_profesionales

- **Tabla Salarios**

Table: salarios						
Fields						
Name	Type	Not Null	Unique	P/K	A/I	Binary
PROFESIONALES_PERSONAS_ID_Persona	int(11)	Not Null		Yes		
Mes	varchar(100)	Not Null		Yes		
Anho	year(4)	Not Null		Yes		
Salario_Bruto	float	Not Null				
Seguridad_Social	float	Not Null				
Seguridad_Social_Empresa	float	Not Null				
IRPF	float	Not Null				

Figura 56. Tabla Salarios

Indices				
Index Name	On Field	Unique	Full Text	Sorting
PRIMARY	`PROFESIONALES_PERSONA S_ID_Persona`, `Mes`, `Anho`	Yes		
fk_SALARIOS_PROFESIONALES	`PROFESIONALES_PERSONA S_ID_Persona`			
SALARIO_PROF_INDEX	`PROFESIONALES_PERSONA S_ID_Persona`			

Figura 57. Índices tabla Salarios

- **Tabla Voluntarios**

Table: voluntarios						
Fields						
Name	Type	Not Null	Unique	P/K	A/I	Binary
PERSONAS_ID_Persona	int(11)	Not Null		Yes		
ID_Voluntario	int(11)	Not Null	Yes			
Tipo	varchar(100)	Not Null				
Titulacion	varchar(100)	Not Null				
Disponibilidad	varchar(100)	Not Null				
Observaciones	longtext					

Figura 58. Tabla Voluntarios

Indices				
Index Name	On Field	Unique	Full Text	Sorting
PRIMARY	`PERSONAS_ID_Persona`	Yes		
VOLUNTARIO_UNICO	`ID_Voluntario`	Yes		
fk_VOLUNTARIOS_PERSONAS	`PERSONAS_ID_Persona`			
VOLUNTARIO_INDEX	`ID_Voluntario`			
PERSONA_VOLUN_INDEX	`PERSONAS_ID_Persona`			

Figura 59. Índices tabla Voluntarios

- **Tabla Histórico gastos voluntarios**

Table: historico_gastos_voluntarios						
Fields						
Name	Type	Not Null	Unique	P/K	A/I	Binary
VOLUNTARIOS_PERSONAS_ID_Persona	int(11)	Not Null		Yes		
Concepto	varchar(100)	Not Null		Yes		
Fecha	date	Not Null		Yes		
Cantidad	float	Not Null				
Observaciones	longtext					

Figura 60. Tabla Histórico_gastos_voluntarios

Indices				
Index Name	On Field	Unique	Full Text	Sorting
PRIMARY	`VOLUNTARIOS_PERSONAS_ID_Persona`, `Concepto`, `Fecha`	Yes		
fk_HISTORICO_GASTOS_VO LUNTARIOS_VOLUNTARIOS	`VOLUNTARIOS_PERSONAS_ID_Persona`			
PERSONA_GASTOS_VOLUN _INDEX	`VOLUNTARIOS_PERSONAS_ID_Persona`			
CONCEPTO_VOLUN_INDEX	`Concepto`			
VOLUNT_CONCEPTO_INDEX	`VOLUNTARIOS_PERSONAS_ID_Persona`, `Concepto`			

Figura 61. Índices tabla Histórico_gastos_voluntarios

- **Tabla Actividades**

Table: actividades						
Fields						
Name	Type	Not Null	Unique	P/K	A/I	Binary
ID_Actividad	int(11)	Not Null		Yes	Yes	
Nombre	varchar(100)	Not Null	Yes			
Comentarios	longtext					

Figura 62. Tabla Actividades

Indices				
Index Name	On Field	Unique	Full Text	Sorting
PRIMARY	`ID_Actividad`	Yes		
ACTIVIDAD_UNICA	`Nombre`	Yes		
NOMBRE_ACT_INDEX	`Nombre`			

Figura 63. Índices tabla Actividades

- **Tabla Periodos actividades**

Table: periodos_actividades						
Fields						
Name	Type	Not Null	Unique	P/K	A/I	Binary
INFORMACION_ACTIVIDADES_ACTIVIDADES_ID_Actividad	int(11)	Not Null		Yes		
INFORMACION_ACTIVIDADES_Curso	varchar(100)	Not Null		Yes		
INFORMACION_ACTIVIDADES_Grupo	varchar(100)	Not Null		Yes		
Fecha_Alta	datetime	Not Null		Yes		
Fecha_Baja	datetime	Not Null		Yes		
Observaciones	longtext					

Figura 64. Tabla Periodos_actividades

Indices				
Index Name	On Field	Unique	Full Text	Sorting
PRIMARY	`Fecha_Alta`, `Fecha_Baja`, `INFORMACION_ACTIVIDADES_ACTIVIDADES_ID_Actividad`, `INFORMACION_ACTIVIDADES_Curso`, `INFORMACION_ACTIVIDADES_Grupo`	Yes		
fk_PERIODOS_ACTIVIDADES_INFORMACION_ACTIVIDADES	`INFORMACION_ACTIVIDADES_ACTIVIDADES_ID_Actividad`, `INFORMACION_ACTIVIDADES_Curso`, `INFORMACION_ACTIVIDADES_Grupo`			
ACT_CURSO_GRUPO_INDEX	`INFORMACION_ACTIVIDADES_ACTIVIDADES_ID_Actividad`, `INFORMACION_ACTIVIDADES_Curso`, `INFORMACION_ACTIVIDADES_Grupo`			

Figura 65. Índices tabla Periodos_actividades

- **Tabla Información actividades**

Table: informacion_actividades						
Fields						
Name	Type	Not Null	Unique	P/K	A/I	Binary
ACTIVIDADES_ID_Actividad	int(11)	Not Null		Yes		
Curso	varchar(100)	Not Null		Yes		
Grupo	varchar(100)	Not Null		Yes		
Lugar	varchar(100)	Not Null				
Horario	varchar(100)	Not Null				
Precio	float					
Responsable	varchar(100)	Not Null				

Figura 66. Tabla Información_actividades

Indices				
Index Name	On Field	Unique	Full Text	Sorting
PRIMARY	`ACTIVIDADES_ID_Actividad`, `Curso`, `Grupo`	Yes		
fk_INFORMACION_ACTIVIDADES_ACTIVIDADES	`ACTIVIDADES_ID_Actividad`			
ACT_CURSO_INDEX	`ACTIVIDADES_ID_Actividad`, `Curso`			

Figura 67. Índices tabla Información_actividades

- **Tabla Usuarios realizan actividades**

Table: usuarios_realizan_actividades						
Fields						
Name	Type	Not Null	Unique	P/K	A/I	Binary
USUARIOS_PERSONAS_ID_Persona	int(11)	Not Null		Yes		
INFORMACION_ACTIVIDADES_ACTIVIDADES_ID_Actividad	int(11)	Not Null		Yes		
INFORMACION_ACTIVIDADES_Curso	varchar(100)	Not Null		Yes		
INFORMACION_ACTIVIDADES_Grupo	varchar(100)	Not Null		Yes		
Fecha_Alta	datetime	Not Null		Yes		
Fecha_Baja	datetime					

Figura 68. Tabla Usuarios_realizan_actividades

Indices				
Index Name	On Field	Unique	Full Text	Sorting
PRIMARY	`USUARIOS_PERSONAS_ID_Persona`, `INFORMACION_ACTIVIDADES_ACTIVIDADES_ID_Actividad`, `INFORMACION_ACTIVIDADES_Curso`, `INFORMACION_ACTIVIDADES_Grupo`, `Fecha_Alta`	Yes		
fk_USUARIOS_has_INFORMACION_ACTIVIDADES_USUARIOS	`USUARIOS_PERSONAS_ID_Persona`			
fk_USUARIOS_has_INFORMACION_ACTIVIDADES_INFORMACION_ACTIVIDADES	`INFORMACION_ACTIVIDADES_ACTIVIDADES_ID_Actividad`, `INFORMACION_ACTIVIDADES_Curso`, `INFORMACION_ACTIVIDADES_Grupo`			
USUARIOS_ACT_CURSO_GRUPO_INDEX	`INFORMACION_ACTIVIDADES_ACTIVIDADES_ID_Actividad`, `INFORMACION_ACTIVIDADES_Curso`, `INFORMACION_ACTIVIDADES_Grupo`			
ACTIVIDADES_USUARIO_INDEX	`USUARIOS_PERSONAS_ID_Persona`			

Figura 69. Índices tabla Usuarios_realizan_actividades

- **Tabla Profesionales imparten actividades**

Table: profesionales_imparten_actividades						
Fields						
Name	Type	Not Null	Unique	P/K	A/I	Binary
PROFESIONALES_PERSONAS_ID_Persona	int(11)	Not Null		Yes		
INFORMACION_ACTIVIDADES_ACTIVIDADES_ID_Actividad	int(11)	Not Null		Yes		
INFORMACION_ACTIVIDADES_Curso	varchar(100)	Not Null		Yes		
INFORMACION_ACTIVIDADES_Grupo	varchar(100)	Not Null		Yes		
Fecha_Alta	datetime	Not Null		Yes		
Fecha_Baja	datetime					

Figura 70. Tabla Profesionales_imparten_actividades

Indices				
Index Name	On Field	Unique	Full Text	Sorting
PRIMARY	`PROFESIONALES_PERSONAS_ID_Persona`, `INFORMACION_ACTIVIDADES_ACTIVIDADES_ID_Actividad`, `INFORMACION_ACTIVIDADES_Curso`, `INFORMACION_ACTIVIDADES_Grupo`, `Fecha_Alta`	Yes		
fk_PROFESIONALES_has_INFORMACION_ACTIVIDADES_PROFESIONALES	`PROFESIONALES_PERSONAS_ID_Persona`			
fk_PROFESIONALES_has_INFO_ACTS_INFO_ACTS	`INFORMACION_ACTIVIDADES_ACTIVIDADES_ID_Actividad`, `INFORMACION_ACTIVIDADES_Curso`, `INFORMACION_ACTIVIDADES_Grupo`			
PROFS_ACT_CURSO_GRUPO_INDEX	`INFORMACION_ACTIVIDADES_ACTIVIDADES_ID_Actividad`, `INFORMACION_ACTIVIDADES_Curso`, `INFORMACION_ACTIVIDADES_Grupo`			
ACTIVIDADES_PROFESIONALES_INDEX	`PROFESIONALES_PERSONAS_ID_Persona`			

Figura 71. Índices tabla Profesionales_imparten_actividades

- **Tabla Voluntarios colaboran actividades**

Table: voluntarios_colaboran_actividades						
Fields						
Name	Type	Not Null	Unique	P/K	A/I	Binary
VOLUNTARIOS_PERSONAS_ID_Persona	int(11)	Not Null		Yes		
INFORMACION_ACTIVIDADES_ACTIVIDADES_ID_Actividad	int(11)	Not Null		Yes		
INFORMACION_ACTIVIDADES_Curso	varchar(100)	Not Null		Yes		
INFORMACION_ACTIVIDADES_Grupo	varchar(100)	Not Null		Yes		
Fecha_Alta	datetime	Not Null		Yes		
Fecha_Baja	datetime					
Comentarios	longtext					

Figura 72. Tabla Voluntarios_colaboran_actividades

Indices				
Index Name	On Field	Unique	Full Text	Sorting
PRIMARY	`VOLUNTARIOS_PERSONAS_ID_Persona`, `INFORMACION_ACTIVIDADES_ACTIVIDADES_ID_Actividad`, `INFORMACION_ACTIVIDADES_Curso`, `INFORMACION_ACTIVIDADES_Grupo`, `Fecha_Alta`	Yes		
fk_VOLUNTARIOS_has_INFORMACION_ACTIVIDADES_VOLUNTARIOS	`VOLUNTARIOS_PERSONAS_ID_Persona`			
fk_VOLUNTARIOS_has_INFORMACION_ACTIVIDADES_ACTS_INFO_ACTS	`INFORMACION_ACTIVIDADES_ACTIVIDADES_ID_Actividad`, `INFORMACION_ACTIVIDADES_Curso`, `INFORMACION_ACTIVIDADES_Grupo`			
VOLUNTARIOS_ACT_CURSO_GRUPO_INDEX	`INFORMACION_ACTIVIDADES_ACTIVIDADES_ID_Actividad`, `INFORMACION_ACTIVIDADES_Curso`, `INFORMACION_ACTIVIDADES_Grupo`			
ACTS_VOLUNTARIO_INDEX	`VOLUNTARIOS_PERSONAS_ID_Persona`			

Figura 73. Índices tabla Voluntarios_colaboran_actividades

- **Tabla Programas**

Table: programas						
Fields						
Name	Type	Not Null	Unique	P/K	A/I	Binary
ID_Programa	int(11)	Not Null		Yes	Yes	
Nombre	varchar(100)	Not Null				
Curso	varchar(100)	Not Null				
Responsable	varchar(100)	Not Null				
Descripcion	longtext					

Figura 74. Tabla Programas

Indices				
Index Name	On Field	Unique	Full Text	Sorting
PRIMARY	`ID_Programa`	Yes		
PROG_CURSO_UNICO	`Nombre`, `Curso`	Yes		
NOMBRE_PROG_INDEX	`Nombre`, `Curso`			

Figura 75. Índices tabla Programas

- **Tabla Actividad pertenece programa**

Table: actividad_pertenece_programa						
Fields						
Name	Type	Not Null	Unique	P/K	A/I	Binary
ACTIVIDADES_ID_Actividad	int(11)	Not Null		Yes		
PROGRAMAS_ID_Programa	int(11)	Not Null		Yes		

Figura 76. Tabla Actividad_pertenece_programa

Indices				
Index Name	On Field	Unique	Full Text	Sorting
PRIMARY	`ACTIVIDADES_ID_Actividad`, `PROGRAMAS_ID_Programa`	Yes		
fk_ACTIVIDADES_has_PROG RAMAS_ACTIVIDADES	`ACTIVIDADES_ID_Actividad`			
fk_ACTIVIDADES_has_PROG RAMAS_PROGRAMAS	`PROGRAMAS_ID_Programa`			
PROGRAMA_ACT_INDEX	`ACTIVIDADES_ID_Actividad`			
ACTS_PROGRAMA_INDEX	`PROGRAMAS_ID_Programa`			

Figura 77. Índices tabla Actividad_pertenece_programa

- **Tabla Instituciones**

Table: instituciones						
Fields						
Name	Type	Not Null	Unique	P/K	A/I	Binary
ID_Institucion	int(11)	Not Null		Yes	Yes	
DIRECCIONES_ID_Direccion	int(11)	Not Null				
Nombre	varchar(100)	Not Null	Yes			
Departamento	varchar(100)					
Telefono	varchar(100)					
E-mail	varchar(100)					
Pagina_Web	varchar(100)					

Figura 78. Tabla Instituciones

Indices				
Index Name	On Field	Unique	Full Text	Sorting
PRIMARY	`ID_Institucion`	Yes		
NOMBRE_INST_UNICO	`Nombre`	Yes		
fk_INSTITUCIONES_DIRECCIONES	`DIRECCIONES_ID_Direccion`			
NOMBRE_INST_INDEX	`Nombre`			

Figura 79. Índices tabla Instituciones

- **Tabla Subvenciones**

Table: subvenciones						
Fields						
Name	Type	Not Null	Unique	P/K	A/I	Binary
Nombre	varchar(100)	Not Null		Yes		
Anho	year(4)	Not Null		Yes		
INSTITUCIONES_ID_Institucion	int(11)	Not Null				
Tipo	varchar(100)	Not Null				
Fecha_Solicitud	date	Not Null				
Presupuestado	float	Not Null				
Solicitado	float	Not Null				
Concedido	float	Not Null				
Estado	varchar(100)	Not Null				

Figura 80. Tabla Subvenciones

Indices				
Index Name	On Field	Unique	Full Text	Sorting
PRIMARY	`Nombre` , `Anho`	Yes		
fk_SUBVENCIONES_INSTITUCIONES	`INSTITUCIONES_ID_Institucion`			
SUBVENCION_INDEX	`Nombre`			
SUBVS_INST_INDEX	`INSTITUCIONES_ID_Institucion`			

Figura 81. Índices tabla Subvenciones

- **Tabla Proveedores**

Table: proveedores						
Fields						
Name	Type	Not Null	Unique	P/K	A/I	Binary
ID_Proveedor	int(11)	Not Null		Yes	Yes	
Nombre	varchar(100)	Not Null				
CIF	varchar(100)	Not Null	Yes			
Persona_Contacto	varchar(100)	Not Null				
Telefono_Contacto	varchar(100)	Not Null				
E-mail_Contacto	varchar(100)					
Pagina_Web	varchar(100)					
Observaciones	longtext					

Figura 82. Tabla Proveedores

Indices				
Index Name	On Field	Unique	Full Text	Sorting
PRIMARY	`ID_Proveedor`	Yes		
CIF_PROV_UNICO	`CIF`	Yes		
NOMBRE_PROV_INDEX	`Nombre`			
CIF_PROV_INDEX	`CIF`			

Figura 83. Índices tabla Proveedores

- **Tabla Facturas**

Table: facturas						
Fields						
Name	Type	Not Null	Unique	P/K	A/I	Binary
ID_Factura	varchar(100)	Not Null		Yes		
PROVEEDORES_ID_Proveedor	int(11)	Not Null				
ID_Factura_Proveedor	varchar(100)					
Fecha	date	Not Null				
Concepto	varchar(100)	Not Null				
Importe	float	Not Null				
Estado	varchar(100)	Not Null				

Figura 84. Tabla Facturas

Indices				
Index Name	On Field	Unique	Full Text	Sorting
PRIMARY	`ID_Factura`	Yes		
fk_FACTURAS_PROVEEDORES	`PROVEEDORES_ID_Proveedor`			
FACTS_PROV_INDEX	`PROVEEDORES_ID_Proveedor`			
FACTURA_INDEX	`ID_Factura`			

Figura 85. Índices tabla Facturas

- **Tabla Líneas factura**

Table: lineas_factura						
Fields						
Name	Type	Not Null	Unique	P/K	A/I	Binary
FACTURAS_ID_Factura	varchar(100)	Not Null		Yes		
ID_Linea_Factura	int(11)	Not Null		Yes		
Cantidad	float	Not Null				
Descripcion	varchar(100)	Not Null				

Figura 86. Tabla Líneas_factura

Indices				
Index Name	On Field	Unique	Full Text	Sorting
PRIMARY	`FACTURAS_ID_Factura`, `ID_Linea_Factura`	Yes		
fk_LINEAS_FACTURA_FACTURAS	`FACTURAS_ID_Factura`			
LINEAS_FACT_INDEX	`FACTURAS_ID_Factura`			

Figura 87. Índices tabla Líneas_factura

- **Tabla Facturas subvenciones**

Table: facturas_subvenciones						
Fields						
Name	Type	Not Null	Unique	P/K	A/I	Binary
FACTURAS_ID_Factura	varchar(100)	Not Null		Yes		
SUBVENCIONES_Nombre	varchar(100)	Not Null		Yes		
SUBVENCIONES_Anho	year(4)	Not Null		Yes		
Fecha	date	Not Null				
Porcentaje	float	Not Null				
Importe	float	Not Null				

Figura 88. Tabla Facturas_subvenciones

Indices				
Index Name	On Field	Unique	Full Text	Sorting
PRIMARY	`FACTURAS_ID_Factura`, `SUBVENCIONES_Nombre`, `SUBVENCIONES_Anho`	Yes		
fk_FACTURAS_has_SUBVENCIONES_FACTURAS	`FACTURAS_ID_Factura`			
FACT_APORTA_SUBV_ANHO_INDEX	`SUBVENCIONES_Nombre`, `SUBVENCIONES_Anho`			
fk_FACTURAS_SUBVENCIONES_SUBVENCIONES	`SUBVENCIONES_Nombre`, `SUBVENCIONES_Anho`			
FACT_SUBV_ANHO_INDEX	`FACTURAS_ID_Factura`, `SUBVENCIONES_Nombre`, `SUBVENCIONES_Anho`			

Figura 89. Índices tabla Facturas_subvenciones

- **Tabla Profesionales subvenciones**

Table: profesionales_subvenciones						
Fields						
Name	Type	Not Null	Unique	P/K	A/I	Binary
PROFESIONALES_PERSONAS_ID_Persona	int(11)	Not Null		Yes		
SUBVENCIONES_Nombre	varchar(100)	Not Null		Yes		
SUBVENCIONES_Anho	year(4)	Not Null		Yes		
Fecha	date	Not Null				
Porcentaje	float	Not Null				
Importe	float	Not Null				

Figura 90. Tabla Profesionales_subvenciones

Indices				
Index Name	On Field	Unique	Full Text	Sorting
PRIMARY	`PROFESIONALES_PERSONAS_ID_Persona`, `SUBVENCIONES_Nombre`, `SUBVENCIONES_Anho`	Yes		
fk_PROFESIONALES_has_SUBVENCIONES_PROFESIONALES	`PROFESIONALES_PERSONAS_ID_Persona`			
fk_PROFESIONALES_has_SUBVENCIONES_SUBVENCIONES	`SUBVENCIONES_Nombre`, `SUBVENCIONES_Anho`			
PROF_SUBV_ANHO_INDEX	`PROFESIONALES_PERSONAS_ID_Persona`, `SUBVENCIONES_Nombre`, `SUBVENCIONES_Anho`			
PROF_PRESENT_SUBV_INDEX	`SUBVENCIONES_Nombre`, `SUBVENCIONES_Anho`			

Figura 91. Índices tabla Profesionales_subvenciones

- **Tabla Direcciones**

Table: direcciones						
Fields						
Name	Type	Not Null	Unique	P/K	A/I	Binary
ID_Direccion	int(11)	Not Null		Yes	Yes	
Calle	varchar(100)	Not Null				
Numero	int(11)					
Escalera	varchar(100)					
Piso	varchar(100)					
Letra	char(1)					
Codigo_Postal	varchar(100)					
Poblacion	varchar(100)	Not Null				
Provincia	varchar(100)	Not Null				

Figura 92. Tabla Direcciones

Indices				
Index Name	On Field	Unique	Full Text	Sorting
PRIMARY	`ID_Direccion`	Yes		
POBLACION_INDEX	`Poblacion`			
CALLE_INDEX	`Calle`			

Figura 93. Índices tabla Direcciones

- **Tabla Perfiles**

Table: perfiles						
Fields						
Name	Type	Not Null	Unique	P/K	A/I	Binary
ID_Perfil	int(11)	Not Null		Yes	Yes	
Nombre	varchar(100)	Not Null	Yes			
Menu_personas	int(11)					
Menu_programas	int(11)					
Menu_actividades	int(11)					
Menu_usuarios_acts	int(11)					
Menu_instituciones	int(11)					
Menu_subvenciones	int(11)					
Menu_proveedores	int(11)					
Menu_facturas	int(11)					
Menu_salarios	int(11)					
Menu_gastos_extras	int(11)					
Menu_acts_extras	int(11)					
Menu_perfiles	int(11)					
Menu_personal	int(11)					
Menu_password	int(11)					

Figura 94. Tabla Perfiles

Indices				
Index Name	On Field	Unique	Full Text	Sorting
PRIMARY	`ID_Perfil`	Yes		
NOMBRE_PERFIL_UNICO	`Nombre`	Yes		
NOMBRE_PERFIL_INDICE	`Nombre`			

Figura 95. Índices tabla Perfiles

- **Tabla Personal**

Table: personal						
Fields						
Name	Type	Not Null	Unique	P/K	A/I	Binary
ID_Personal	int(11)	Not Null		Yes	Yes	
PERFILES_ID_Perfil	int(11)	Not Null				
Nombre	varchar(100)					
Login	varchar(100)		Yes			
Password	varchar(100)					

Figura 96. Tabla Personal

Indices				
Index Name	On Field	Unique	Full Text	Sorting
PRIMARY	`ID_Personal`	Yes		
LOGIN_UNICO	`Login`	Yes		
fk_PERSONAL_PERFILES	`PERFILES_ID_Perfil`			
NOMBRE_PERSONAL_INDICE	`Nombre`			
LOGIN_PERSONAL_INDICE	`Login`			
PERFIL_PERSONAL_INDICE	`PERFILES_ID_Perfil`			

Figura 97. Índices tabla Personal

- **Tabla Dominios**

Table: dominios						
Fields						
Name	Type	Not Null	Unique	P/K	A/I	Binary
ID_Dominio	varchar(100)	Not Null		Yes		
Descripcion	varchar(100)		Yes			

Figura 98. Tabla Dominios

Indices				
Index Name	On Field	Unique	Full Text	Sorting
PRIMARY	`ID_Dominio`	Yes		
DOMINIO_UNICO	`Descripcion`	Yes		
DESC_DOMINIO_INDEX	`Descripcion`			

Figura 99. Índices tabla Dominios

- **Tabla Descripciones**

Table: descripciones						
Fields						
Name	Type	Not Null	Unique	P/K	A/I	Binary
ID_Descripcion	int(11)	Not Null		Yes	Yes	
Descripcion	varchar(100)	Not Null	Yes			
DOMINIOS_ID_Dominio	varchar(100)	Not Null				

Figura 100. Tabla Descripciones

Indices				
Index Name	On Field	Unique	Full	Sorting
PRIMARY	`ID_Descripcion`	Yes		
DESC_DOMINIO_UNICO	`Descripcion`	Yes		
fk_DESCRIPCIONES_DOMINIOS	`DOMINIOS_ID_Dominio`			

Figura 101. Índices tabla Descripciones

- **Tabla Tarifas gastos**

Table: tarifas_gastos						
Fields						
Name	Type	Not Null	Unique	P/K	A/I	Binary
ID_Tarifa_Gastos	int(11)	Not Null		Yes	Yes	
Concepto	varchar(100)	Not Null				
Anho	year(4)	Not Null				
Precio	float	Not Null				

Figura 102. Tabla Tarifas_gastos

Indices				
Index Name	On Field	Unique	Full Text	Sorting
PRIMARY	`ID_Tarifa_Gastos`	Yes		
CONCEPTO_ANHO	`Concepto` , `Anho`	Yes		
TARIFA_CONCEPTO_ANHO_INDEX	`Concepto` , `Anho`			

Figura 103. Índices tabla Tarifas_gastos

- **Tabla Cuotas**

Table: cuotas						
Fields						
Name	Type	Not Null	Unique	P/K	A/I	Binary
Anho	year(4)	Not Null		Yes		
Concepto	varchar(100)	Not Null		Yes		
Importe	float	Not Null				

Figura 104. Tabla Cuotas

Indices				
Index Name	On Field	Unique	Full Text	Sorting
PRIMARY	`Anho`, `Concepto`	Yes		
CUOTA_CONCEPTO_ANHO_INDEX	`Anho`, `Concepto`			

Figura 105. Índices tabla Cuotas

- **Tabla Índices**

Table: indices						
Fields						
Name	Type	Not Null	Unique	P/K	A/I	Binary
Ind_socio	int(11)					
Ind_usuario	int(11)					
Ind_voluntario	int(11)					
Ind_profesional	int(11)					

Figura 106. Tabla Índices

3.3.- Diseño de la interfaz

Para dar por finalizado el apartado dedicado al diseño de un sistema informático de gestión Web para la Asociación Navarra de Síndrome de Down se va realizar a continuación una pequeña descripción de la interfaz gráfica elaborada con el objetivo de satisfacer los requisitos de apariencia y GUI presentados en el apartado 2.2.1 del presente documento.

El concepto de interfaz gráfica de usuario o GUI (Graphic User Interface) puede definirse como el conjunto de formas y métodos que posibilitan la interacción de un sistema con los usuarios empleando para ello formas gráficas (botones, ventanas, fuentes, etc.) e imágenes que representen las distintas funciones, acciones e información disponibles en la aplicación de manera que se posibilite una interacción amigable entre el usuario y el sistema informático. [GUIDef] [GUIDefWiki]

Como puede deducirse de la definición que se acaba de realizar, el aspecto fundamental de una buena interfaz gráfica es que facilite de la mejor manera posible la interacción entre el usuario y el sistema informático que se le presenta como respuesta a sus necesidades. Por este motivo, el diseño de la interfaz gráfica debe estar condicionado por el tipo de usuarios que van a hacer uso de la aplicación informática a la que corresponde dicha interfaz. En el caso que nos ocupa, y como ya se ha comentado en alguna otra ocasión, el perfil de usuario al que va dirigida la herramienta responde a un perfil con conocimientos básicos de informática y no familiarizado con el uso de aplicaciones Web para el desempeño de sus funciones laborales por lo que la interfaz diseñada debe ser lo más sencilla e intuitiva posible.

Atendiendo al tipo de usuario al que va dirigida la aplicación Web se ha optado por realizar una interfaz gráfica lo más simple y funcional posible que facilite en la medida de lo posible la labor diaria del personal de la Asociación Navarra de Síndrome de Down. Así, y como se irá viendo a lo largo de las diferentes figuras que se presentaran en este apartado, se ha elaborado una interfaz gráfica compuesta únicamente por dos colores básicos coincidentes con los colores del logotipo de la Asociación con el objetivo de no distraer la atención del usuario hacia aspectos no relacionados con la funcionalidad propia de la Aplicación.

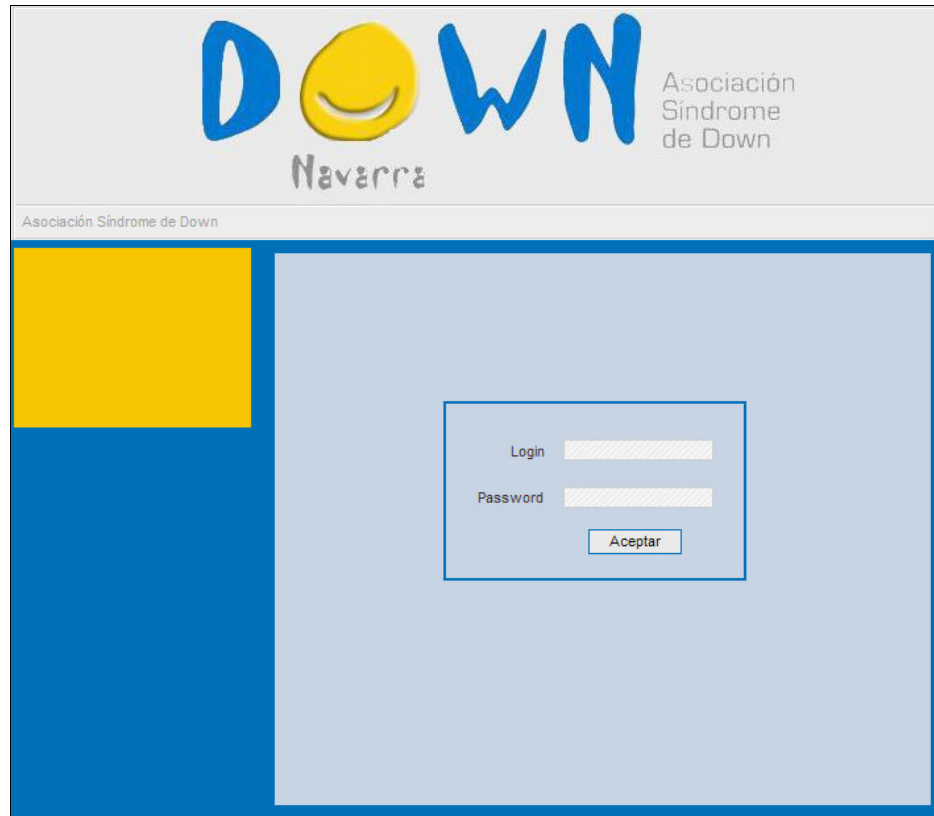


Figura 107. Pantalla de acceso a la aplicación

Como puede observarse en la figura anterior, la pantalla de la aplicación se compone de tres partes bien diferenciadas. En la parte superior, y a modo identificativo, se muestra el logotipo de la Asociación, en la parte izquierda se presenta la sección en la que, una vez el usuario se haya logado, se mostrará el menú con los diferentes apartados que responden a la funcionalidad de la aplicación (el control de acceso a cada una de las opciones del menú se realizará internamente) y finalmente, en la parte central de la pantalla y ocupando la mayor parte de la misma se muestra el área de trabajo a través de la que el usuario podrá realizar las diferentes acciones permitidas por la aplicación. Esta misma estructura será la que se mantenga en todas y cada una de las pantallas que componen la herramienta de manera que se facilite la familiarización del usuario con la aplicación y se evite el hecho de que en cada pantalla se tenga que dedicar tiempo a la búsqueda de las opciones que permiten desempeñar la funcionalidad deseada.

Otro de los aspectos importantes a considerar a la hora de realizar el diseño de una aplicación informática es el relacionado con la nomenclatura empleada en la misma. Se debe tener en cuenta que los textos empleados tanto en las diferentes secciones del menú como en los botones de acción y en todos los campos y mensajes mostrados en la herramienta deben ser textos breves y concisos. Es decir, los textos deben ser siempre lo más descriptivos posibles de manera que faciliten al usuario conocer la acción que se llevará a cabo al pulsar un determinado botón o sección del menú y permitan la fácil comprensión de los diferentes mensajes mostrados por la aplicación.

Siguiendo con la descripción de la interfaz gráfica desarrollada para la aplicación de gestión Web para la Asociación Navarra de Síndrome de Down se van a presentar a continuación las páginas diseñadas para llevar a cabo las diferentes funcionalidades proporcionadas por la aplicación. En todas ellas, y como se podrá observar a lo largo de las diferentes imágenes que se van a mostrar, se siguen cumpliendo los criterios de estructuración de pantalla y nomenclatura presentados hasta el momento, además de los nuevos criterios que se van a ir definiendo a partir de este momento.

Asociación Síndrome de Down Navarra

Asociación Síndrome de Down > Personal

Menú

- Personas
- Programas
- Actividades
- Usuarios-Actividades
- Entidades
- Subvenciones
- Proveedores
- Facturas
- Salarios
- Gastos extraordinarios
- Actividades extraordinarias
- Perfiles
- Personal
- Cambio password

Menú >> Personal

Nombre:

Login:

Perfil:

Listado Personal

Nombre	Login	Perfil
Superadministrador	SUPERADMINISTRADOR	Superadministrador
Maria Jiménez	MJIMENEZ	Gestor
Alberto Sanz	ASANZ	Monitor
Carmen Marchena	CMARCHENA	Informático
Javier Amatriain	JAMATRIAIN	Contable

<< Comienzo < Anterior Siguiente > Fin >>

Figura 108. Pantalla listado datos

La Figura 108 que se acaba de presentar muestra el diseño que se va a seguir en todas las pantallas existentes para mostrar los datos de distintos tipos gestionados por la herramienta (en el caso de la figura mostrada se presenta el listado de personal de la Asociación que va a poder hacer uso de la aplicación Web). El primer aspecto que debe comentarse al observar la figura anterior es el relacionado con el hecho de que en este momento el menú de la aplicación es perfectamente visible para el usuario puesto que éste ya se ha logado correctamente en la aplicación. Como se observa, en dicho menú se muestran todas las opciones existentes sin realizar ningún tipo de filtro puesto que éste se realizará internamente a través del código de la aplicación, de manera que se permita al usuario conocer en todo momento el alcance funcional de la aplicación. Así mismo, se puede observar que, para cumplir con el objetivo de facilitar al máximo al usuario el empleo de la aplicación, la pantalla central se ha estructurado en tres partes diferenciadas, estructura que se mantendrá en el resto de pantallas del mismo tipo existentes en la aplicación.

En la parte superior se muestra la opción del menú a la que ha accedido el usuario con el fin de que éste conozca fácilmente a qué elemento de la aplicación afectará la acción que se esté llevando a cabo, así como el botón de desconexión que se mostrará en el mismo lugar en todas las pantallas de la aplicación de manera que se permita al usuario abandonar la herramienta en el momento en que lo desee. El hecho de que el botón de desconectar se muestre en la parte superior derecha de la pantalla, al igual que se verá más adelante en relación a los botones de acción existentes en la pantalla, coincide con la razón de que esa es la zona que más fácilmente identifica el usuario y, por lo tanto, la zona en la que se han situado los botones más importantes de la aplicación.

Justo debajo de la opción del menú accedida se presenta una pequeña subsección a través de la cual el usuario podrá llevar a cabo las diferentes acciones posibles para la opción del menú accedida. En este momento es importante indicar que la estructura de dicha subsección será la misma en todas las pantallas de este tipo mostrando en la parte izquierda el listado de campos por los que es posible realizar una búsqueda y en la parte derecha los botones de acción correspondientes a las posibles acciones a llevar a cabo. En lo referente a los botones de acción cabe indicar que su situación en la parte superior derecha de la pantalla no es casual, sino que se sitúan en dicha zona debido a que, al ser los botones más importantes de la pantalla, se muestran en la zona de la misma que el usuario visualiza con mayor facilidad sin necesidad de tener que emplear los desplazamientos por la pantalla. Del mismo modo, la ordenación de los botones mostrando en primer lugar el botón de “Crear” y en segundo lugar el de “Buscar” tampoco es casual sino que responde al hecho de que, para evitar que el usuario pierda tiempo buscando los botones de acción, éstos siempre se van a ordenar mostrando en primer lugar los botones correspondientes a las acciones más importantes existentes en ese momento en la pantalla.

Finalmente, en la parte central de la pantalla se mostrará el listado de datos existentes para la sección accedida que cumplan el criterio de búsqueda introducido, en el caso de que se hubiera realizado una búsqueda, o todos los existentes si no se ha realizado búsqueda alguna. En lo referente al diseño de este listado se debe indicar que siempre mostrarán un máximo de cinco registros por página con el objetivo de evitar que el usuario se vea abrumado por la cantidad de información existente y sienta reparo a la hora de emplear la herramienta además de por evitar que el usuario tenga que emplear de manera excesiva el desplazamiento vertical para poder encontrar la

información deseada (en el caso de que existan más de cinco registros a mostrar la aplicación mostrará la posibilidad de desplazarse por las diferentes páginas de resultados). Así mismo, y atendiendo igualmente al criterio de que el usuario no se sienta desbordado por la información existente, el listado no mostrará todos los campos existentes en la aplicación para cierto tipo de datos sino que mostrará solo aquellos que permitan identificar de manera unívoca la información mostrada. Por último, en el diseño de la interfaz de este tipo de páginas se debe tener en cuenta que todos los campos por los que se pueda realizar una búsqueda en el listado deben mostrarse en dicho listado en el mismo orden en que aparecen en la sección de búsqueda. Con este requisito de diseño se pretende no confundir al usuario con la información mostrada y que éste pueda comprobar de manera directa que los registros mostrados en el listado coinciden con el criterio de búsqueda introducido en la sección correspondiente de la pantalla.

Para continuar con la descripción de la interfaz gráfica de usuario empleada en la aplicación se presentan a continuación dos figuras correspondientes a la creación y modificación de datos a través de las cuales se pretende explicar el criterio seguido en el empleo de los botones de acción. El primer criterio a tener en cuenta, y que puede observarse al contemplar las Figuras 109 y 110 es, además del hecho de que la nomenclatura de los botones debe ser breve y concisa indicando claramente la funcionalidad de los mismos, que en cada pantalla se muestran únicamente aquellos botones que realmente son útiles en cada ocasión para ofrecer al usuario solo aquellas acciones que realmente pueden llevarse a cabo y no confundirle con botones inapropiados en un determinado momento. Así, puede observarse que en la Figura 109 correspondiente a la inserción de datos no se muestra el botón de borrar, cosa que si ocurre en la Figura 110 relativa a la modificación de un registro ya existente. Otro de los aspectos que se ha tenido en cuenta en la realización de la aplicación Web es el relativo a la situación de los botones de acción dentro de la pantalla debido a que se considera que éstos deben situarse en el lugar apropiado de manera que se guíe al usuario hasta su utilización. Por ello, en el caso de las dos figuras siguientes los botones de acción se muestran en la parte inferior de la pantalla justo debajo de los campos que el usuario debe insertar o modificar de manera obligatoria para poder llevar a cabo la creación o modificación de un registro. Es decir, los botones de acción se sitúan en la parte de la pantalla a la que el usuario va a llegar de manera directa guiado por el funcionamiento de la aplicación. Finalmente, en lo relativo al diseño de los botones, se debe indicar que la ordenación de los mismos se realiza mostrando más a la derecha el botón de mayor importancia en la pantalla y avanzando hacia la izquierda según disminuye la importancia de la funcionalidad representada por el botón en cuestión. Dicha ordenación se realiza de este modo porque se considera que los usuarios identifican más fácilmente los botones situados a la derecha, motivo por el que en dicha posición se muestra el botón de guardar seguido del botón de borrar, mientras que el botón de volver, al no ser un botón correspondiente a la funcionalidad propia de la aplicación, se muestra en un segundo lugar en la parte izquierda de la pantalla.

DOWN Navarra Asociación Síndrome de Down

Asociación Síndrome de Down ▸ Personal

Menú

- Personas
- Programas
- Actividades
- Usuarios-Actividades
- Entidades
- Subvenciones
- Proveedores
- Facturas
- Salarios
- Gastos extraordinarios
- Actividades extraordinarias
- Perfiles
- Personal
- Cambio password

Menú >> Personal Desconectar

Nombre

Login

Password

Perfil

Figura 109. Pantalla inserción datos

DOWN Navarra Asociación Síndrome de Down

Asociación Síndrome de Down ▸ Personal

Menú

- Personas
- Programas
- Actividades
- Usuarios-Actividades
- Entidades
- Subvenciones
- Proveedores
- Facturas
- Salarios
- Gastos extraordinarios
- Actividades extraordinarias
- Perfiles
- Personal
- Cambio password

Menú >> Personal Desconectar

Nombre

Login

Password

Perfil

Figura 110. Pantalla modificación datos

Con el objetivo de tratar el apartado relativo al diseño de la interfaz gráfica de la aplicación en todos los tipos de pantallas existentes en la herramienta se van a mostrar a continuación dos figuras correspondientes a los mensajes típicos, tanto de error como de información, que se muestran en la aplicación para dar a conocer al usuario el motivo por el que se ha producido un determinado error o el resultado que ha producido en la aplicación el correcto uso de la misma.

Como puede observarse, en la parte superior izquierda de las Figuras 111 y 112, al igual que en el resto de pantallas de la herramienta, se muestra la opción del menú accedida por el usuario de manera que éste pueda identificar de forma fácil y directa a qué elemento de la aplicación corresponde el mensaje de error o información que se está visualizando. Del mismo modo, en la parte derecha de ambas figuras se muestra el botón de desconexión para permitir al usuario abandonar la aplicación en el momento en que lo considere oportuno. Además de estos dos aspectos, al contemplar las dos figuras siguientes vemos que se sigue cumpliendo el aspecto relativo a que las pantallas y mensajes deben ser lo más simples y concisos posibles motivo por el que ambas pantallas muestran únicamente un título que indica si el mensaje es de error o información y un breve y sencillo texto explicando al usuario el motivo del mensaje, además del botón de Volver que permite al usuario abandonar la pantalla actual.

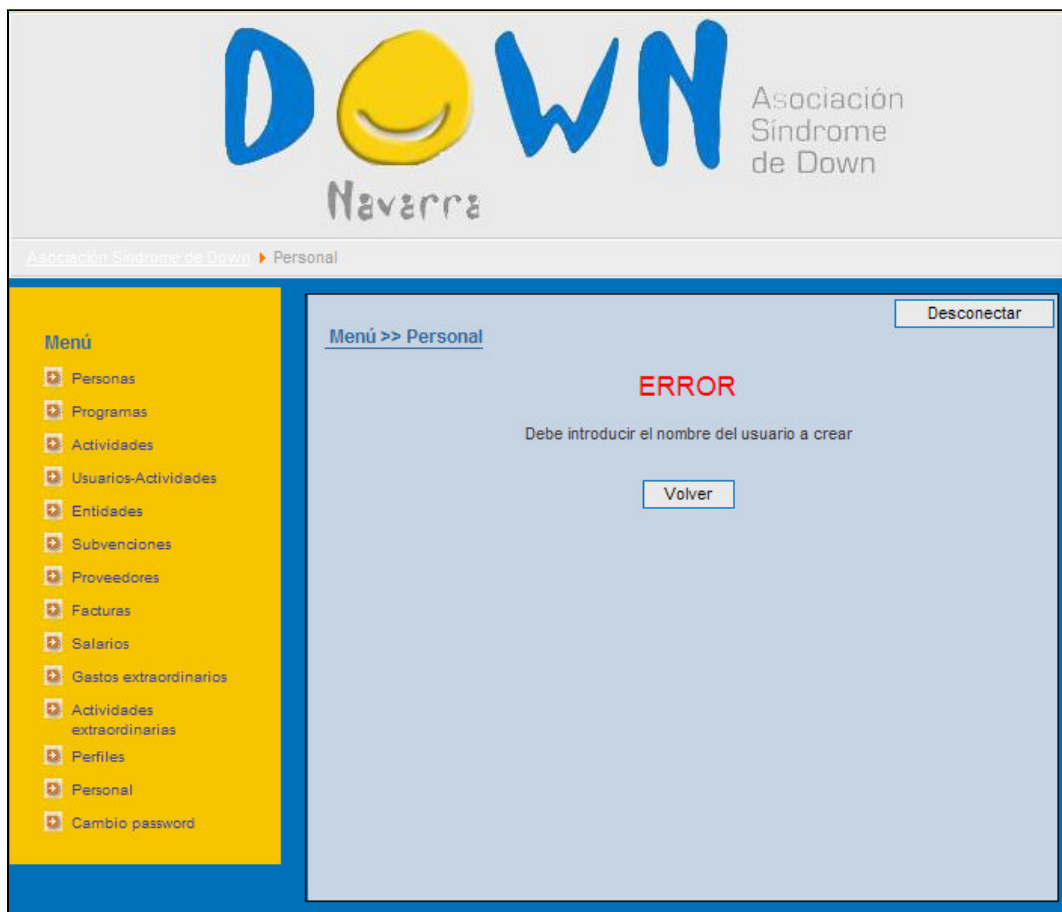


Figura 111. Pantalla mensaje de error

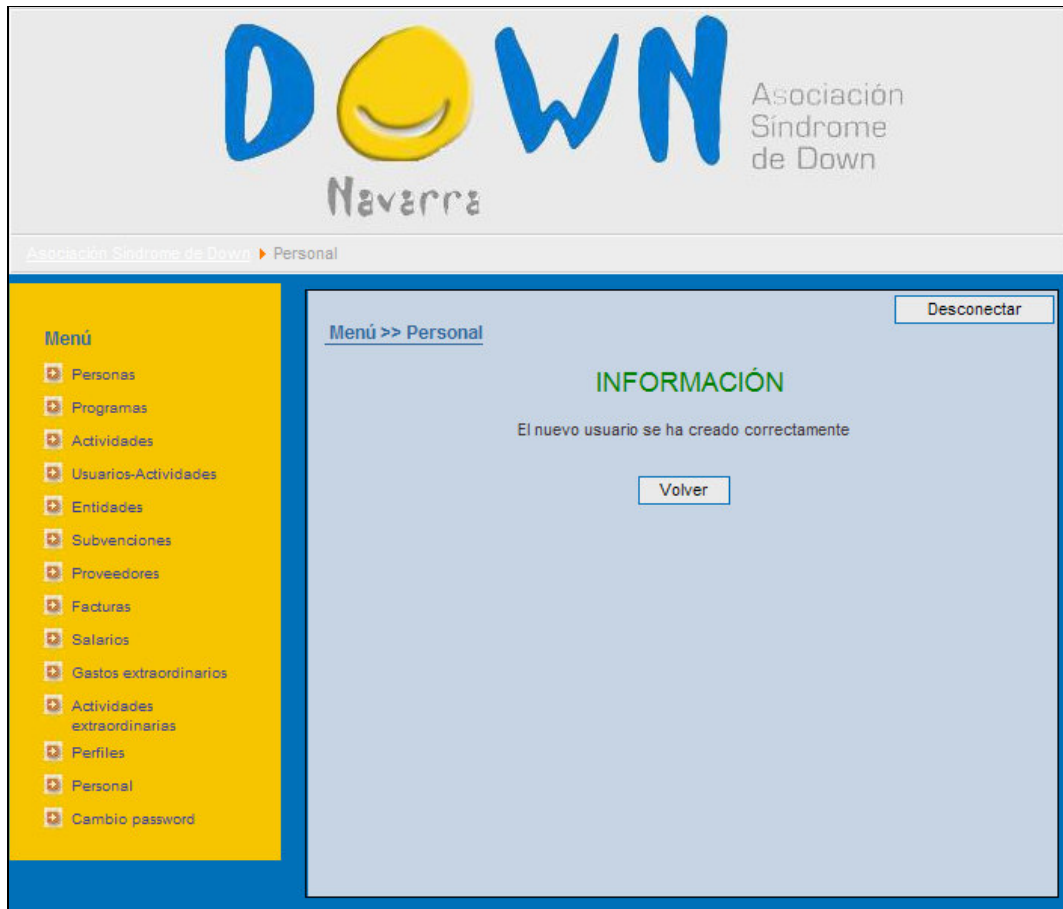


Figura 112. Pantalla mensaje de información

Para dar por finalizado el apartado dedicado al diseño de la interfaz gráfica de la herramienta se puede indicar que el aspecto fundamental que se ha tenido en cuenta a la hora de llevar a cabo esta tarea ha sido el de facilitar en todo momento el trabajo del usuario evitando introducir elementos que pudieran distraer su atención del objetivo fundamental de la aplicación y manteniendo en todo momento la estructura y criterios empleados en las diferentes pantallas de manera que, una vez que el usuario se haya familiarizado con la herramienta, su utilización sea homogénea y coherente en todo momento. Por ello, los aspectos más importantes del diseño han sido los de mantener siempre la misma estructuración en las pantallas, la misma situación, ordenación y nomenclatura en los botones de acción, evitar mostrar al usuario demasiada información al mismo tiempo, indicar en todo momento la opción del menú accedida, emplear mensajes breves y concisos evitando mostrar elementos que no aporten utilidad al usuario en cada momento, todo ello encaminado a disponer de una herramienta sencilla e intuitiva para el usuario.

Llegados a este punto, y a título informativo, se considera importante indicar que para presentar el apartado del diseño de la interfaz gráfica de usuario se han empleado las pantallas relativas a la opción “Personal” de la aplicación, pero que los mismos criterios presentados para dicha opción se mantienen en todas las pantallas existentes para el resto de opciones disponibles en el menú de la herramienta.

Capítulo 4. Implementación

Llegados a este punto de la memoria descriptiva de la aplicación desarrollada, debe estar bien claro que la funcionalidad exigida a la herramienta en forma de requisitos funcionales consiste en que sea capaz de gestionar fácil y eficazmente la información manejada en la Asociación Navarra de Síndrome de Down, tanto en lo que a aspectos directamente relacionados con la actividad de la Asociación se refiere (alumnos, profesores, actividades, programas, etc.) como a aspectos relativos a la gestión de los usuarios que podrán hacer uso de la herramienta. En el primer capítulo de la presente memoria se realizó una pequeña introducción a la Asociación Navarra de Síndrome de Down presentado su situación informática actual así como el objetivo que se pretende alcanzar con la realización del presente proyecto. Los capítulos segundo y tercero se ocuparon de llevar a cabo el análisis de los requisitos a cumplir por la aplicación a desarrollar, así como de presentar el diseño de la solución propuesta. Finalmente, este cuarto capítulo se va a centrar en la presentación del procedimiento de programación seguido en la implementación del código fuente desarrollado basándonos para ello en la presentación del código concreto implementado para la funcionalidad de gestión de personal de la Asociación con acceso a la herramienta.

4.1.- Elementos necesarios para la implementación

Antes de centrarnos en la presentación del código fuente implementado, se va a realizar una pequeña reseña sobre los diferentes elementos empleados para poder llevar a cabo la citada implementación. Dichos elementos, como ya se ha comentado tanto en la presentación de los requisitos no funcionales de la herramienta como en la arquitectura del sistema, son el gestor de contenidos Joomla junto con su extensión FacileForms y el lenguaje de programación PHP, además de una base de datos MySQL en la que quedará almacenado el código fuente implementado.

4.1.1.- Gestor de contenidos Joomla

Joomla es un Sistema de Gestión de Contenidos (CMS) robusto, personalizable y escalable que permite crear sitios Web y aplicaciones en línea de gran alcance, elegantes, dinámicas e interactivas. Por su diseño, facilidad de uso, potencia y flexibilidad así como por sus enormes posibilidades de ampliación, Joomla se está convirtiendo en el sistema de publicación Web más popular de entre todos los disponibles. Se trata además de una aplicación de código abierto disponible libremente para cualquiera que desee utilizarla programada en lenguaje PHP bajo una licencia GPL, que utiliza una base de datos MySQL para almacenar el contenido y los parámetros de configuración del sitio [WebEmp] [TutoJoomla] [EduJoomla].

Un CMS o Content Management System es un programa que permite crear una estructura de soporte (framework) para la creación y administración de documentos y otros contenidos, principalmente en páginas Web, de un modo cooperativo [CMSWiki]. Dicho de otra forma, es una herramienta que permite a un editor crear, clasificar y publicar cualquier tipo de información en una página Web [CMS]. Consiste en una interfaz que controla una o varias bases de datos donde se aloja el contenido, la información o los datos del sitio. El sistema permite manejar de manera independiente el contenido y el diseño de manera que así es posible gestionar el contenido y darle en cualquier momento un diseño distinto al sitio sin tener que darle un nuevo formato al contenido, además de permitir la fácil y controlada publicación en el sitio de varios editores [CMSWiki].

Entre los elementos básicos del Gestor de Contenidos o CMS Joomla encontramos los siguientes [TutoJoomla]:

- **Plantillas.** Las plantillas o templates y sus archivos asociados proporcionan el aspecto visual y el manejo del sitio Web y se mantienen separados del contenido del sitio. Dichas plantillas se almacenan en la base de datos MySQL existente asociada al sitio Web. La instalación estándar de Joomla incluye dos plantillas preinstaladas, pero existen multitud de páginas Web desde donde poder disponer de nuevas plantillas.
- **Componentes.** Los componentes son elementos del núcleo de Joomla con una funcionalidad determinada que se muestran en el cuerpo principal de la plantilla del sitio Web. La instalación estándar de Joomla incluye los siguientes componentes por defecto: Banners, Contactos, Noticias Externas, Encuestas y Enlaces Web, pero al igual que ocurría con el caso de las plantillas es posible disponer e instalar componentes adicionales en Joomla como se verá a continuación al presentar el componente FacileForms empleado para la realización de los formularios de la aplicación Web objeto del proyecto.
- **Módulos.** Los módulos amplían las posibilidades de Joomla proporcionando nueva funcionalidad al software. Un módulo es un pequeño artículo de contenido que puede mostrarse en cualquier parte que permita la plantilla. La instalación estándar de Joomla incluye, entre otros, los siguientes módulos: Menú Principal, Menú Superior, Noticias Externas, Contador de Accesos, etc.
- **Mambots.** Un mambot es una pequeña función orientada a una tarea que intercepta cierto tipo de contenido y lo manipula de algún modo.

Para finalizar con la presentación del Gestor de Contenidos Joomla se debe indicar que ofrece las mismas ventajas que otros gestores de contenidos en la medida en que hace posible convertir una Web estática en un completo portal con diferentes funcionalidades y características dinámicas e interactivas. Entre dichas características pueden encontrarse las citadas a continuación [EduJoomla]:

- Joomla es un sistema administrado, es decir, existe uno o varios administradores encargados de realizar, desde el panel de control, todas las operaciones necesarias para el mantenimiento del sitio Web, incluidas las relativas a la instalación de nuevos componentes y módulos en el sistema.
- La gestión de usuarios de Joomla permite asignar diferentes niveles de uso y administración a distintos usuarios lo que facilita la participación de varios autores en el desarrollo y mantenimiento del sitio Web.
- Joomla facilita la edición y organización de los contenidos del sitio Web gracias a la posibilidad de mostrar tanto páginas de contenido estático para la información más estable y permanente como artículos agrupados en secciones y categorías temáticas.
- Joomla facilita la edición de artículos o páginas de contenido gracias a la existencia de un editor visual con una barra de herramientas semejante a la de los procesadores de textos con la que se puede dar formato al contenido, añadir imágenes, etc.
- Joomla ofrece independencia entre el diseño y el contenido. La estética del sitio Web se basa en plantillas o templates prediseñadas que permiten modificar en cualquier momento el aspecto visual del sitio sin tener que dar de nuevo formato al contenido.

Hasta el momento se han presentado tanto los elementos como las características básicas del Gestor de Contenidos Joomla a través de las cuales es posible construir rápida y fácilmente diferentes tipos de sitios Web, pero en ocasiones las necesidades de las empresas y organizaciones van más allá de lo que es posible proporcionar gracias a los elementos básicos de Joomla. En estos casos, la potente estructura de soporte de Joomla y el hecho de que esté basado en PHP y MySQL hacen posible que los desarrolladores pueden diseñar y construir funcionalidades específicas para una aplicación concreta gracias al empleo de nuevos componentes que es posible instalar en Joomla y a la implementación de código específico en PHP. La situación que se acaba de presentar es la existente en el caso de la aplicación que nos ocupa en este proyecto puesto que la funcionalidad básica de Joomla no permitía desarrollar una herramienta a medida que cubriera las necesidades solicitadas por la Asociación Navarra de Síndrome de Down. Por este motivo, en la realización del presente proyecto ha sido necesaria la instalación en Joomla del componente FacileForms, así como la programación a medida de código fuente en PHP.

4.1.1.1.- Componente FacileForms

Como ya es bien sabido, la necesidad existente en la Asociación Navarra de Síndrome de Down y planteada como objeto del presente proyecto pasa por la realización de una aplicación Web a medida capaz de satisfacer la carencia de gestión informática en el proceso administrativo de la Asociación. Dicha necesidad, como ya se ha planteado en el apartado anterior, no puede satisfacerse con los elementos que el Gestor de Contenidos Joomla ofrece en su versión estándar, puesto que se requiere una herramienta capaz de crear formularios a través de los cuales el usuario pueda interactuar con el sistema para llevar a cabo las labores administrativas derivadas de la actividad de la Asociación. Por ello, se hace necesaria la instalación de un componente adicional de Joomla capaz de proporcionar los citados formularios para la interacción entre el usuario y la herramienta.

El componente seleccionado para la creación tanto de los formularios como del código fuente necesario para la implementación de la aplicación para la Asociación Navarra de Síndrome de Down ha sido FacileForms, componente obtenido de la página oficial de extensiones de Joomla en castellano [Ext.Joomla] e instalado según el proceso indicado en la referencia [FormFacileForms]. Se trata de una herramienta que permite generar formularios para el Gestor de Contenidos Joomla, caracterizada por crear tablas propias dentro de la base de datos del Gestor de Contenidos instalado en las que se almacena de manera directa todos los elementos generados en el formulario. La creación de formularios con FacileForms es directa y sencilla puesto que basta con seleccionar el elemento que se quiere incluir, establecer sus propiedades y escoger su ubicación dentro del formulario [FacileFormsJoomla]. Además de la creación directa de formularios, FacileForms permite la introducción de código fuente propio en PHP a través del cual se realiza la implementación de la funcionalidad específica requerida por la aplicación a desarrollar, tal y como se verá en el segundo apartado de este cuarto capítulo al presentar el procedimiento de programación seguido para implementar la aplicación para la Asociación Navarra de Síndrome de Down.

4.1.2.- Lenguaje de programación PHP

Antes de comenzar con la explicación a través de un ejemplo del procedimiento de programación seguido para implementar la funcionalidad de la aplicación objeto del presente proyecto, se va a realizar una pequeña introducción al lenguaje de programación PHP, que es el lenguaje empleado para realizar la implementación.

PHP, acrónimo de Hypertext Preprocessor, es un lenguaje de programación de script interpretado (no se compila para conseguir código máquina sino que existe un intérprete que lee el código y se encarga de ejecutar las instrucciones que contiene dicho código) [Int.PHP] de alto nivel que puede ser incrustado en páginas HTML [ManualPHP] y usado especialmente para crear contenido Web dinámico y aplicaciones para servidores [Def.PHP].

A diferencia de Java o Javascript que se ejecutan en el navegador, la interpretación y ejecución del código PHP se realiza en el servidor, es decir, el programa PHP es ejecutado en el servidor generando como resultado una página HTML que se envía al cliente. De esta manera el cliente recibirá los resultados de ejecutar el código PHP sin ninguna posibilidad de determinar qué código ha producido el resultado recibido. [Def.PHP] [ManualPHP] [Carac.PHP]

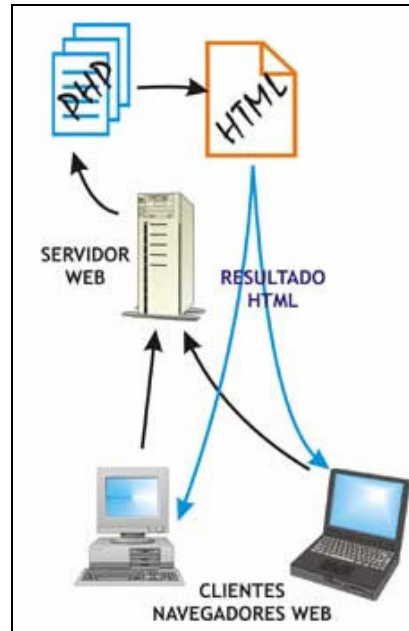


Figura 113. Funcionamiento PHP

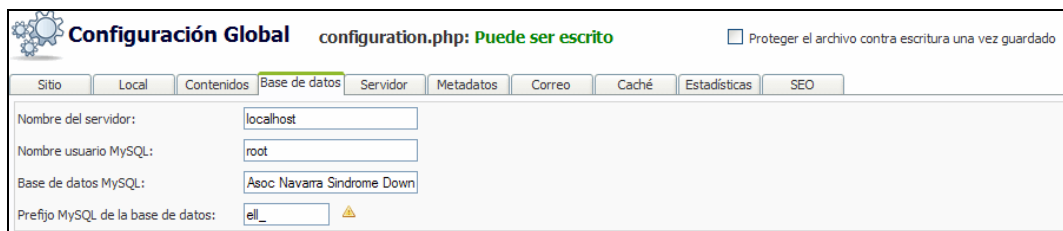
Después de haber realizado una pequeña introducción al lenguaje de programación PHP se van a presentar a continuación algunas de sus principales características [HistoriaPHP] [Carac.PHP]:

- Es un lenguaje libre distribuido de forma gratuita bajo una licencia abierta lo que lo convierte en una herramienta de fácil acceso ideal para la creación de páginas Web dinámicas.
- Es un lenguaje multiplataforma, rápido y de fácil aprendizaje.
- Es un lenguaje que dispone de capacidad de conexión con la mayoría de los manejadores de bases de datos existentes en la actualidad: MySQL, PostgreSQL y Oracle entre otras.
- Es independiente del navegador, es decir, al ser un lenguaje que se ejecuta en el servidor no es necesario que el navegador lo soporte, basta con que lo haga el servidor en el que están alojadas las páginas PHP.
- Permite las técnicas de programación orientada a objetos.
- No requiere definición de tipos de variables.
- Dispone de manejo de excepciones desde PHP 5.

4.2.- Procedimiento de programación y ejemplo funcionalidad “Gestión de Personal”

Una vez realizada en el apartado anterior del presente capítulo una pequeña introducción a los elementos necesarios para poder llevar a cabo la implementación de la aplicación a desarrollar para la Asociación Navarra de Síndrome de Down, es momento en este apartado de presentar en detalle el procedimiento de programación empleado para obtener la funcionalidad deseada por la citada aplicación. Debido a que la presentación del ya citado procedimiento de programación que se ha seguido puede ser un poco complicada de explicar en sí misma se va a realizar a través de la exposición del código fuente programado para la implementación de la funcionalidad de la gestión del personal de la Asociación con acceso a la herramienta.

Antes de comenzar con la explicación del procedimiento de programación seguido se debe indicar que al instalar el Gestor de Contenidos Joomla es muy importante indicar la base de datos contra la que se trabajará (base de datos que debe estar previamente creada en el servidor MySQL). Es decir, se debe indicar la base de datos en la que se almacenarán tanto los formularios y elementos creados como el código implementado y toda la información de la Asociación que se gestione a través de la herramienta desarrollada. Para comprobar que la configuración de la base de datos es correcta basta con acceder a la pestaña “Base de datos” existente en la Configuración Global de Joomla a la que se accede desde la opción “Sitio >> Configuración Global” una vez que se encuentre dentro de la administración del Gestor de Contenidos.



The screenshot shows the Joomla! Global Configuration interface, specifically the 'Base de datos' (Database) tab. The title bar indicates 'configuration.php: Puede ser escrito' and a checkbox for 'Proteger el archivo contra escritura una vez guardado'. The tabs at the top are: Sitio, Local, Contenidos, Base de datos (selected), Servidor, Metadatos, Correo, Caché, Estadísticas, and SEO. The form contains the following fields:

Nombre del servidor:	localhost
Nombre usuario MySQL:	root
Base de datos MySQL:	Asoc Navarra Síndrome Down
Prefijo MySQL de la base de datos:	ell_

Figura 114. Configuración base de datos Joomla

Como se observa en la imagen anterior, toda la información derivada del empleo de la aplicación se almacenará en la base de datos “Asoc Navarra Síndrome Down” alojada en el servidor localhost. Además, en dicha base de datos Joomla creará sus propias tablas con el prefijo “ell_” donde almacenará toda la información relacionada con los formularios, elementos y código creados para poder dar lugar a la herramienta implementada.

Una vez comprobado que la configuración de la base de datos es correcta, nos centramos en la explicación del procedimiento de programación objeto del apartado actual. La primera acción a llevar a cabo según el procedimiento seguido es la de construir o componer los formularios que contendrán las diferentes páginas que constituirán la aplicación que visualizará el usuario. Es decir, el primer paso consiste en la creación de la interfaz a través de la que el usuario interactuará con la herramienta desarrollada. En este momento es importante indicar que para poder llevar una mejor organización de las diferentes pantallas de la aplicación, se ha creado un formulario independiente para cada una de las opciones de menú con que contará la herramienta. Antes de presentar el proceso seguido para la creación de los formularios y la composición de las pantallas que los constituyen, se debe indicar que la estructuración

que se va a seguir para la creación de las pantallas de la aplicación va a ser la de mostrar una primera pantalla en la que se muestre el listado de elementos existentes junto con las opciones de crear y buscar y una segunda pantalla correspondiente a la creación, modificación y borrado de un elemento concreto. Dicho esto, procedemos a relatar el proceso de creación de las citadas pantallas o formularios.

Como ya se ha indicado en el apartado anterior, el componente empleado para la creación de los formularios es FacileForms, componente que, como se verá a continuación, facilita en gran medida la composición de las pantallas puesto que sus elementos pueden crearse y configurarse directamente a través de la interfaz que el componente proporciona para ello. Una vez dentro de la gestión de formularios de FacileForms el usuario administrador podrá crear tantos formularios como considere necesarios a través del botón “Nuevo” que existe para ello. Al crear un nuevo formulario FacileForms muestra una ventana como la que se presenta en la Figura 115 en la que se deberá realizar la configuración inicial del formulario indicando aspectos tales como el nombre o el tamaño del formulario.

The screenshot shows the 'Facile Forms - Editar formulario' window. It has a tabbed interface with 'Piezas de form' selected. The main area contains various configuration options:

- Ajustes** (selected):
 - Título: Personal
 - Paquete:
 - Nombre: Crear_personal
 - Clase CSS para <div>: content_outline
 - Clase CSS para <form>:
 - Ordenar: 3 (Personal)
 - Publicado: ☐ No ☒ Sí
 - Modo de ejecución: Cualquiera
 - Anchura: 500 px
 - Altura: Fijado 600 px
 - Modo previo: Sobreexpuesto
 - Registro en base de datos: Valores no vacíos
 - Correo-e de notificación: Dirección por defecto
 - Informe: Valores no vacíos
 - XML Adjunto: No
- Scripts**
- Piezas de form** (selected):
 - Antes del Formulario**: Tipo: ☒ Ninguno ☐ Librería ☐ Personalizado
 - Después del Formulario**: Tipo: ☒ Ninguno ☐ Librería ☐ Personalizado
- Enviar piezas**

At the bottom, there is a description field with the text: 'Formulario para la creación de personal (usuarios que podrán acceder a la herramienta)'. At the bottom right, there are 'Guardar' (Save) and 'Cancelar' (Cancel) buttons.

Figura 115. Configuración inicial formulario

Además de los ajustes iniciales, como se observa en la figura anterior, es posible indicar el código que se ejecutará en los diferentes estados por los que pasa el formulario (script de inicialización, script enviado, antes del formulario, después del formulario, antes de enviar y después de enviar). Por el momento, no se va a indicar ningún código adicional sino que solamente se realiza la configuración de los ajustes básicos del formulario.

Después de haber creado el formulario se visualiza una pantalla como la mostrada en la siguiente figura donde el administrador podrá ir creando y componiendo las distintas páginas que constituirán el formulario a través de los distintos botones existentes en la barra de herramientas. Al pulsar el botón “Nuevo” FacileForms abrirá una ventana emergente como la mostrada en la Figura 116 para que el usuario seleccione el nuevo tipo de elemento que desea introducir en la pantalla. Una vez seleccionado se le presentará una pantalla similar a la mostrada en la Figura 115 donde se podrá realizar la configuración de los ajustes básicos del elemento y donde se podrá introducir el código que se desee en cada una de las opciones disponibles. Al igual que en el caso de la creación del formulario, por el momento no se introduce código alguno sino que solamente se realiza la configuración básica del elemento. Además, a través de los diferentes botones existentes en la barra de herramientas de la Figura 116 se podrán crear nuevos elementos copia de otros existentes en otra página, eliminar elementos, crear nuevas páginas, etc.

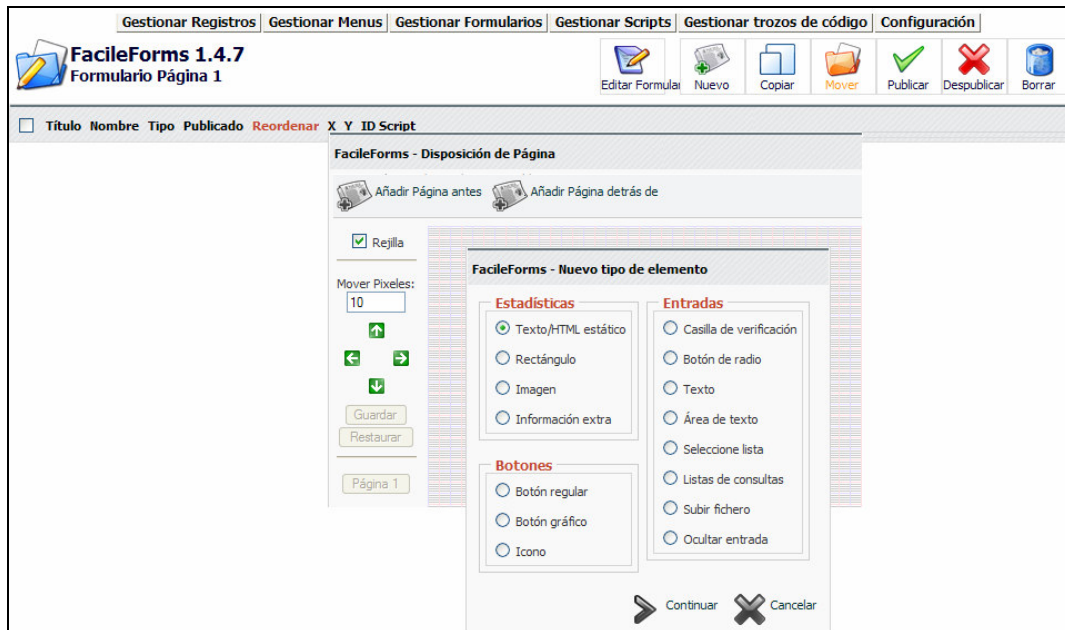


Figura 116. Creación y composición de pantallas

Repitiendo el proceso de creación de elementos que se acaba de describir el usuario acabará por obtener la disposición de la pantalla deseada que, en el caso de la primera página del formulario de gestión de personal de la Asociación con acceso a la aplicación, es la que se muestra en la figura siguiente.

FacileForms - Disposición de Página

Publicado Reordenar X Y ID Script

Añadir Página antes Añadir Página detrás de Mover Página Borrar Página

☒ Regilla

Mover Píxeles: 10

Guardar Restaurar

Página 1
 Página 2
 Página 3
 Página 4
 Página 5
 Página 6
 Página 7
 Página 8
 Página 9
 Página 10
 Página 11
 Página 12
 Página 13
 Página 14
 Página 15
 Página 16
 Página 17
 Página 18
 Página 19
 Página 20

Menú >> Personal

Desconectar

Crear

Buscar

Seleccione perfil

Listado Personal

Nombre	Login	Perfil
Superadministrador	SUPERADMINISTRADOR	Superadministrador
Maria Jiménez	MJIMENEZ	Voluntario
Alberto Sanz	ASANZ	Monitor
Carmen Marchena	CMARCHENA	Informático
Javier Amatriain	JAMATRIAIN	Contable

<< Comienzo < Anterior Siguiente > Fin >>

Figura 117. Pantalla inicial formulario gestión de personal

Como se puede observar en la figura anterior, el formulario correspondiente a la gestión de personal con acceso a la herramienta cuenta con veinte páginas construidas del mismo modo que se acaba de presentar. Las dos primeras páginas corresponden a las pantallas de listado, creación, modificación, búsqueda y borrado de personal existente mientras que el resto de páginas existentes son meras páginas informativas sobre el éxito o error en la realización de las diferentes acciones existentes en el formulario.

Una vez visto el procedimiento para la creación de formularios y pantallas dentro de éstos a través del componente FacileForms de Joomla, vamos a centrarnos a partir de ahora en los aspectos meramente funcionales, es decir, vamos a centrarnos en explicar tanto el funcionamiento interno de FacileForms como el código fuente implementado dentro de los diferentes elementos que FacileForms deja abiertos para que el usuario programe a su criterio con el fin de obtener la funcionalidad deseada para la aplicación que deba implementar. Comenzamos indicando que FacileForms permite ejecutar código asociado a los formularios en seis puntos o estados diferentes del ciclo de vida del propio formulario:

- **Script de inicialización.** El código introducido en esta sección se ejecutará en el momento de creación del formulario.
- **Script enviado.** El código introducido en esta sección se ejecutará en el momento en que se produzca el envío del formulario, entre la ejecución del código existente en el estado “Antes de enviar” y el código existente en el estado “Después de enviar”.

- **Antes del formulario.** El código introducido en esta sección se ejecutará antes de que se carguen todos y cada uno de los elementos que constituyen el formulario.
- **Después del formulario.** El código introducido en esta sección se ejecutará después de que se carguen todos y cada uno de los elementos que constituyen el formulario.
- **Antes de enviar.** El código introducido en esta sección se ejecutará justo antes de que se produzca el envío del formulario debido a la acción asociada a un determinado botón.
- **Después de enviar.** El código introducido en esta sección se ejecutará justo después de que se produzca el envío del formulario debido a la acción asociada a un determinado botón.

En todos y cada uno de los puntos o estados existentes en el formulario que se acaban de mencionar el usuario programador dispone de tres posibilidades a la hora de introducir código fuente. Así, puede decidir no introducir código alguno asociado al evento en cuestión, puede decidir ejecutar código propio de FacileForms o puede incluir código fuente propio y específico para la funcionalidad deseada.

Además de las diferentes posibilidades que FacileForms ofrece para introducir código directamente asociado al formulario, cada uno de los distintos elementos que pueden introducirse para constituir una pantalla dentro del formulario también disponen de distintas alternativas en las que el usuario programador puede introducir código propio. Así, la mayoría de estos elementos disponen de las alternativas para introducir código al entrar en el formulario, al entrar en la pantalla, al ejecutar una determinada acción (clic, desenfoco, cambiar, enfoque o seleccionar) sobre el elemento en cuestión o al realizar una validación sobre el valor introducido en el elemento, mientras que los elementos de tipo botón solamente disponen de la posibilidad de introducir código asociado al momento en el que se produzca la acción vinculada al propio botón. Al igual que ocurría en el caso del código asociado al propio formulario, el código introducido en cada uno de los elementos que constituyen una pantalla puede ser código propio de FacileForms o código desarrollado por el usuario programador. Por último, cada uno de los elementos que es posible introducir en las distintas pantallas de un formulario pueden llevar asociado un valor por defecto que será el que se muestre cuando se cargue la pantalla en cuestión, valor que posteriormente podrá ser modificado en función de las acciones derivadas de la funcionalidad de la propia herramienta.

Siguiendo con la introducción de las diferentes alternativas que FacileForms ofrece para introducir código fuente, y una vez presentados todos aquellos puntos en los que es posible introducirlo, se considera conveniente indicar a modo de resumen el flujo que una aplicación implementada con FacileForms sigue en cuanto a ejecución de código se refiere. En primer lugar se ejecuta el código introducido en el evento “Script de inicialización” seguido del existente en el evento “Antes del formulario”. En tercer lugar se ejecutaría el código introducido en el script de inicialización de todos y cada uno de los elementos existentes en la página actual bien sea de entrada de formulario o de entrada de página y a continuación se ejecutaría el código correspondiente al evento “Después del formulario”. Finalmente, y siempre en función de las acciones que realice el usuario, se ejecutaría el código asociado a la acción de pulsar un botón en cuyo caso a

continuación se desencadenaría la ejecución de los códigos de formulario asociados a los eventos de “Antes de enviar”, “Script enviado” y “Después de enviar”. De esta manera, en el código propio del botón solamente se ejecutará la acción de enviar el formulario (submit), en el código del evento “Antes de enviar” se realizarán las comprobaciones y acciones sobre la base de datos pertinentes y finalmente se enviará el formulario ejecutando a continuación, en el caso de que las hubiera, las acciones existentes en el evento “Después de enviar”.

Después de haber presentado de manera teórica el funcionamiento de FacileForms en cuanto a ejecución de código se refiere se va a explicar e introducir a continuación el código implementado para la funcionalidad de gestión de personal con el objetivo de clarificar lo máximo posible el funcionamiento de FacileForms.

En el caso concreto de la funcionalidad de gestión de personal no existe código que deba ejecutarse al inicializar el formulario ni antes de que se carguen los diferentes elementos que lo constituyen, es decir, en este caso toda la ejecución se va a realizar a partir de la inicialización de los elementos. Como se puede observar en la Figura 108, la pantalla principal de la gestión de personal cuenta con cuatro campos y tres botones principales de los cuales solamente el campo de búsqueda de perfil y el campo de listado de personal cuentan con código a ejecutar en su inicialización. Es decir, al cargar la pantalla inicial de la gestión de personal solamente se ejecutarán las dos porciones de código que se van a mostrar a continuación y que, en ambos casos, se ejecutan con el objetivo de mostrar valor por defecto en ambos campos.

El motivo por el que el campo de búsqueda de perfil cuenta con código en su inicialización es debido a que, como uno de los campos por los que es posible realizar una búsqueda de personal existente en la Asociación es a través de su perfil, es necesario que el campo de búsqueda de perfil sea un campo de tipo lista de selección que muestre inicialmente como posibles valores a seleccionar el listado de perfiles existentes en la aplicación para lo cual se ejecuta el siguiente código.

```
<?php
/*Recuperamos el listado de perfiles existentes en la base de
datos*/
$perfilesExistentesCrear= ff_select(
    "select ".
    "Nombre, ID_Perfil ".
    "from perfiles ".
    "order by ID_Perfil "
);

/*Hacemos que en la lista de valores aparezca seleccionada por
defecto la opción de "Seleccione perfil"*/
$opcionesCrear = "1;Seleccione perfil;0";

/*Recorremos el array obtenido de la consulta a la BBDD
obteniendo el nombre de cada uno de los perfiles existentes
e incluyendo dicho perfil como una posible opción de selección en
la lista de valores que se muestre no seleccionada.*/
foreach ($perfilesExistentesCrear as $perfilesECrear) {
    $nombrePerfilCrear = $perfilesECrear->Nombre;
    $idPerfilCrear = $perfilesECrear->ID_Perfil;
    $opcionesCrear .= "\n";
    $opcionesCrear .= "0;$nombrePerfilCrear;$idPerfilCrear";
}

//Devolvemos la lista de valores creada
return $opcionesCrear;
?>
```

Por su parte, el motivo por el que es necesario que el campo correspondiente al listado de valores cuente con código de inicialización es debido a que cuando el usuario accede a la pantalla de gestión de personal debe visualizar un listado con todos los registros de personal existentes en ese momento en la aplicación. Para ello, el citado campo cuenta con el código que se muestra a continuación en el que se realiza una consulta a la base de datos para recuperar todos los registros de personal existentes y donde se construye en HTML el listado a través del cual dichos registros serán devueltos al usuario.

```
$this->execPieceByName('ff_InitLib');

/*El código mostrado a continuación se ejecutará cada vez que el
usuario acceda a la página del formulario y cada vez que realice una
búsqueda sobre los registros existentes en la página. Por ello, es
necesario diferenciar un caso de otro y para ello se recuperan los
valores de nombre, login y perfil que pueden llegar a través de la
URL. Si estos valores son vacíos es debido a que no se ha introducido
criterio de búsqueda o a que se está accediendo de nuevo a la página.
En cualquiera de los dos casos se deberá realizar una consulta a la
BBDD que devuelva todos los registros de personal existentes. Si
alguno de estos valores no es vacío significa que se ha realizado una
búsqueda sobre la página por lo que la consulta a la BBDD deberá
realizarse filtrando por los criterios correspondientes*/
```

```
//Recuperamos los valores de las variables pasadas por la URL
if(isset($_GET['ff_nombre_personal'])){
    /*Nombre de personal pasado por la URL. Recuperamos su valor para
    incluirlo en la consulta a la BBDD*/
    $nombreBuscarPers = $_GET['ff_nombre_personal'];
    if(isset($_GET['ff_login'])){
        /*Login de personal pasado por la URL. Recuperamos su valor para
        incluirlo en la consulta a la BBDD*/
        $loginBuscarPers = $_GET['ff_login'];
        if(isset($_GET['ff_perfil'])){
            /*Perfil de personal pasado por la URL. Recuperamos su valor
            para incluirlo en la consulta a la BBDD*/
            $perfilBuscarPers = $_GET['ff_perfil'];
        }
        else{
            //Perfil de personal no pasado por la URL
            $perfilBuscarPers = "";
        }
    }
    else{
        //Login de personal no pasado por la URL
        $loginBuscarPers = "";
        if(isset($_GET['ff_perfil'])){
            /*Perfil de personal pasado por la URL. Recuperamos su valor
            para incluirlo en la consulta a la BBDD*/
            $perfilBuscarPers = $_GET['ff_perfil'];
        }
        else{
            //Perfil de personal no pasado por la URL
            $perfilBuscarPers = "";
        }
    }
}
else{
    //Nombre de personal no pasado por la URL
    $nombreBuscarPers = "";
    if(isset($_GET['ff_login'])){
        /*Login de personal pasado por la URL. Recuperamos su valor para
        incluirlo en la consulta a la BBDD*/
        $loginBuscarPers = $_GET['ff_login'];
        if(isset($_GET['ff_perfil'])){
            /*Perfil de personal pasado por la URL. Recuperamos su valor
            para incluirlo en la consulta a la BBDD*/
            $perfilBuscarPers = $_GET['ff_perfil'];
        }
        else{
            //Perfil de personal no pasado por la URL
            $perfilBuscarPers = "";
        }
    }
    else{
        //Login de personal no pasado por la URL
        $loginBuscarPers = "";
    }
}
```



```
if(isset($_GET['ff_perfil'])){
    /*Perfil de personal pasado por la URL. Recuperamos su valor
    para incluirlo en la consulta a la BBDD*/
    $perfilBuscarPers = $_GET['ff_perfil'];
}
else{
    //Perfil de personal no pasado por la URL
    $perfilBuscarPers = "";
}
}
}

/*Si todas las variables recuperadas de la URL son vacías significa
que no se ha introducido criterio de búsqueda o que se ha accedido de
nuevo a la página. En ambos casos se deben mostrar todos los
registros de personal existentes en la BBDD por lo que se realiza una
consulta sin incluir ningún filtro de selección*/
if($nombreBuscarPers == " && $loginBuscarPers == " && $perfilBuscarPers == 0){
    $rows = ff_select("
        select p.ID_Personal as Personal,per.Nombre as Perfil,
            p.Nombre as NombrePers,p.Login as LoginPers,
            p.Password as Pass
        from personal p,perfiles per
        where p.PERFILES_ID_Perfil = per.ID_Perfil"
    );
}

/*Si alguna de las variables recuperadas de la URL no es vacía
significa que se ha introducido un criterio de búsqueda y por lo
tanto, solo deben mostrarse al usuario aquellos registros de personal
existentes en la BBDD que cumplan la búsqueda introducida por lo que
la consulta se realizará filtrando en cada caso por los criterios de
búsqueda introducidos. Para evitar que las búsquedas sean exactas
antes de realizar la consulta se concatena a la variable de búsqueda
el carácter comodín "%" por delante y por detrás de manera que la
consulta devuelva todos los registros que contengan el criterio de
búsqueda*/
else{
    if($nombreBuscarPers != ""){
        //Nombre de personal pasado por la URL
        $nuevaVbleNombre = "%".$nombreBuscarPers."%";

        if($loginBuscarPers != ""){
            //Login de personal pasado por la URL
            $nuevaVbleLogin = "%".$loginBuscarPers."%";
            if($perfilBuscarPers != 0){
                /*Perfil de personal pasado por la URL.
                Ejecutamos la consulta de recuperación de personal filtrando
                por nombre, login y perfil*/
                $rows = ff_select("
                    select p.ID_Personal as Personal,per.Nombre as Perfil,
                        p.Nombre as NombrePers,p.Login as LoginPers,
                        p.Password as Pass
                    from personal p,perfiles per
```

```

        where p.PERFILES_ID_Perfil = per.ID_Perfil and
              p.nombre like '$nuevaVbleNombre' and
              p.Login like '$nuevaVbleLogin' and
              p.PERFILES_ID_Perfil like '$perfilBuscarPers'"

    );
}
else{
    /*Perfil de personal no pasado por la URL.
    Ejecutamos la consulta de recuperación de personal filtrando
    por nombre y login*/
    $rows = ff_select("
        select p.ID_Personal as Personal,per.Nombre as Perfil,
              p.Nombre as NombrePers,p.Login as LoginPers,
              p.Password as Pass
        from personal p,perfiles per
        where p.PERFILES_ID_Perfil = per.ID_Perfil and
              p.nombre like '$nuevaVbleNombre' and
              p.Login like '$nuevaVbleLogin'"

    );
}
}
else{
    //Login de personal no pasado por la URL.
    if($perfilBuscarPers != 0){
        /*Perfil de personal pasado por la URL.
        Ejecutamos la consulta de recuperación de personal filtrando
        por nombre y perfil*/
        $rows = ff_select("
            select p.ID_Personal as Personal,per.Nombre as Perfil,
                  p.Nombre as NombrePers,p.Login as LoginPers,
                  p.Password as Pass
            from personal p,perfiles per
            where p.PERFILES_ID_Perfil = per.ID_Perfil and
                  p.nombre like '$nuevaVbleNombre' and
                  p.PERFILES_ID_Perfil like '$perfilBuscarPers'"

        );
    }
    else{
        /*Perfil de personal no pasado por la URL.
        Ejecutamos la consulta de recuperación de personal filtrando
        por nombre*/
        $rows = ff_select("
            select p.ID_Personal as Personal,per.Nombre as Perfil,
                  p.Nombre as NombrePers,p.Login as LoginPers,
                  p.Password as Pass
            from personal p,perfiles per
            where p.PERFILES_ID_Perfil = per.ID_Perfil and
                  p.nombre like '$nuevaVbleNombre'"

        );
    }
}
}
}
}

```

```
else{
    //Nombre de personal no pasado por la URL.
    if($loginBuscarPers != ""){
        //Login de personal pasado por la URL.
        $nuevaVbleLog = "%".$loginBuscarPers."%";
        if($perfilBuscarPers != 0){
            /*Perfil de personal pasado por la URL.
            Ejecutamos la consulta de recuperación de personal filtrando
            por login y perfil*/
            $rows = ff_select("
                select p.ID_Personal as Personal,per.Nombre as Perfil,
                    p.Nombre as NombrePers,p.Login as LoginPers,
                    p.Password as Pass
                from personal p,perfiles per
                where p.PERFILES_ID_Perfil = per.ID_Perfil and
                    p.Login like '$nuevaVbleLog' and
                    p.PERFILES_ID_Perfil like '$perfilBuscarPers'"
            );
        }
    }
    else{
        /*Perfil de personal no pasado por la URL.
        Ejecutamos la consulta de recuperación de personal filtrando
        por login*/
        $rows = ff_select("
            select p.ID_Personal as Personal,per.Nombre as Perfil,
                p.Nombre as NombrePers,p.Login as LoginPers,
                p.Password as Pass
            from personal p,perfiles per
            where p.PERFILES_ID_Perfil = per.ID_Perfil and
                p.Login like '$nuevaVbleLog'"
        );
    }
}
else{
    //Login de personal no pasado por la URL.
    if($perfilBuscarPers != 0){
        /*Perfil de personal pasado por la URL.
        Ejecutamos la consulta de recuperación de personal filtrando
        por perfil*/
        $rows = ff_select("
            select p.ID_Personal as Personal,per.Nombre as Perfil,
                p.Nombre as NombrePers,p.Login as LoginPers,
                p.Password as Pass
            from personal p,perfiles per
            where p.PERFILES_ID_Perfil = per.ID_Perfil and
                p.PERFILES_ID_Perfil like '$perfilBuscarPers'"
        );
    }
}
}
```

Una vez visto y explicado el código de inicialización correspondiente al elemento de listado de personal, se va a mostrar a continuación el código del mismo elemento que permite devolver al usuario los registros obtenidos en la consulta en formato tabla mostrando cada registro como un enlace a través del cual se accederá al detalle del personal en cuestión. Para ello, el elemento listado de personal cuenta, como se observa en la siguiente figura, de una pestaña “Columnas” en la que se definirán los campos de la consulta que deben devolverse al usuario estableciendo el formato de vuelta a través de HTML.

FacileForms - Editar Elemento Listas de consultas

Ajustes Consulta **Columnas**

Nuevo Copiar Borrar

<input type="checkbox"/> Título	Nombre	Cabecera	Atributos	Datos	Atributos	Anchura	Reordenar
<input type="checkbox"/> Listado Personal	data	Centrado Arriba	Izquierda Arriba				

FacileForms - Columna de lista de Consultas

Guardar Cancelar

Cabecera

Título: Listado Personal

Clase CSS para <th>:

<th> Span: 1

Atributos: Centrado Amba Ninguno

Datos

Nombre: data

Clase CSS para <td(1)>:

Clase CSS para <td(2)>:

Anchura: px

Atributos: Izquierda Amba Ninguno

Valor: [4] [12] [20]

```
<?php
//create custom table
return
'<table border="1" width="100%">'
```

Guardar Cancelar

Figura 118. Composición tabla listado de personal

Al contemplar la imagen anterior deducimos que el listado que muestra el resultado de la consulta ejecutada va a mostrar como título “Listado Personal” y que va a obtener sus datos de la variable “data” que es la variable en la que FacileForms almacena el resultado de la consulta ejecutada. La composición de la tabla en la que los datos son devueltos al usuario se realiza en el campo “Valor” introduciendo una tabla HTML dentro del código PHP tal y como se observa en el código que se muestra a continuación.

```

/*Devolvemos el listado de personal en forma de una tabla que
contará, para cada uno de los registros a devolver, de una fila en la
que se indicará los nombres de los campos a mostrar (Nombre, Login y
Perfil) y de otra fila en la que se mostrarán los valores para cada
uno de los campos citados. Además, estos valores se mostrarán en
forma de enlace de navegación a la segunda página del formulario
pasando como parámetro a dicha página toda la información recuperada.
Con esto se consigue que, cada vez que el usuario pulse sobre uno de
los registros de la tabla, se acceda al detalle de dicho registro.*/
<?php
return
'<table border="1" width="100%">'.
'<tr>
  <td width="50%" align="center"><b>Nombre</b></td>
  <td width="50%" align="center"><b>Login</b></td>
  <td width="50%" align="center"><b>Perfil</b></td>
</tr>'.
"<tr>
  <td>
    <a href='{ mossite }/index.php?option=com_facileforms&
      ff_name=Crear_personal&ff_param_view=$row->Personal&
      ff_nombre_personal=$row->NombrePers&ff_login=$row->LoginPers&
      ff_perfil=$row->Perfil&ff_Password=$row->Pass&
      ff_page=2\">$row->NombrePers</a>
  </td>
  <td>
    <a href='{ mossite }/index.php?option=com_facileforms&
      ff_name=Crear_personal&ff_param_view=$row->Personal&
      ff_nombre_personal=$row->NombrePers&ff_login=$row->LoginPers&
      ff_perfil=$row->Perfil&ff_Password=$row->Pass&
      ff_page=2\">$row->LoginPers</a>
  </td>
  <td>
    <a href='{ mossite }/index.php?option=com_facileforms&
      ff_name=Crear_personal&ff_param_view=$row->Personal&
      ff_nombre_personal=$row->NombrePers&ff_login=$row->LoginPers&
      ff_perfil=$row->Perfil&ff_Password=$row->Pass&
      ff_page=2\">$row->Perfil</a>
  </td>
</tr>".
'</table><br>';
?>

```

Con el código de inicialización del campo de búsqueda de perfil y del elemento correspondiente al listado de perfiles que se acaba de presentar se da por finalizada la parte correspondiente a la ejecución del código de inicialización de elementos existentes en la primera página del formulario. Según el flujo de ejecución de una aplicación implementada con FacileForms, en este momento debería ejecutarse el código asociado al evento “Después del formulario”, pero como en el caso de la funcionalidad que nos ocupa dicho evento no contiene código, el siguiente código a ejecutar será el asociado a los diferentes botones existentes en la página de listado de personal de la Asociación. Como se puede observar en la Figura 108, dicha página cuenta, además de con la

posibilidad de acceder al detalle de un personal concreto, con tres botones de acción que corresponden a las acciones de crear un nuevo personal, realizar una búsqueda entre el personal existente y llevar a cabo la desconexión del usuario de la aplicación. Procedemos a presentar y explicar a continuación tanto el código asociado a dichos botones como el asociado a la acción correspondiente en el evento “Antes de enviar”-del formulario que se ejecuta como consecuencia de éste.

En el caso de que el usuario pulse el botón para crear un nuevo personal en la Asociación, la acción principal desencadenada por dicho evento es la de mostrar al usuario una nueva página en la que pueda llevar a cabo la nueva creación. Además de esta acción principal el código del botón “Crear” realizará otras acciones adicionales que, tal y como se muestra en el código siguiente, consisten en limpiar los campos de la página de creación en los que el usuario deberá introducir la información del personal a crear.

```
function ff_Btn_crear_personal_action(element, action){
    switch (action) {
        case 'click':
            /*Debido a que la página de creación de un nuevo personal es la
            misma que la de modificación, el botón de borrar el personal
            debe ocultarse cuando se va a crear uno nuevo.
            La función ff_getElementByName permite recuperar un elemento a
            través de su nombre.*/
            ff_getElementByName('btn_borrar_personal').style.display = 'none';

            /*Recuperamos los elementos existentes en la página de creación
            para que el usuario introduzca la información del nuevo perfil
            a crear y les asignamos a todos ellos el valor vacío (al campo
            perfil le asignamos el valor “Selecione perfil”) de manera
            que cuando el usuario acceda a crear un personal no aparezca
            ningún valor por defecto en dichos campos*/
            ff_getElementByName('Valor_nombre_personal').value = "";
            ff_getElementByName('Valor_login').value = "";
            ff_getElementByName('Valor_password').value = "";
            ff_getElementByName('Valor_perfil').value = 0;

            /*A la hora de guardar la información introducida por el
            usuario para el personal es importante saber si la información
            a guardar corresponde a la creación de un nuevo personal o a
            la modificación de uno ya existente. Por este motivo, cuando
            se pulsa el botón de “Crear” ponemos a uno el valor de la
            variable “Vble_oculta_crear_modificar” que nos indica que se
            está creando un nuevo personal*/
            ff_getElementByName('Vble_oculta_crear_modificar').value = 1;

            /*Redireccionamos a la siguiente página existente en el
            formulario que se corresponde con la página de creación de un
            nuevo personal*/
            ff_nextpage();
            break;
        default:;
    }
}
```

Antes de centrarnos en el código relacionado con los elementos existentes en la página del formulario dedicada a la creación de un nuevo personal se va a presentar el código ejecutado cuando se accede al detalle de un personal ya creado puesto que, debido a que la página de destino es la misma en ambos casos, la presentación de su código asociado se realizará de manera conjunta para ambas funcionalidades. El código de inicialización a ejecutar, en el caso de que se acceda a la página de creación o modificación de un personal para visualizar el detalle de un personal ya existente, será el asociado a los campos de nombre, login y password del personal. En los tres casos el código a ejecutar es idéntico y consistente en recuperar el valor del parámetro correspondiente obtenido a través de la URL del enlace accedido y mostrar dicho valor como valor por defecto en el campo asociado, motivo por el cual solo se presenta el código de inicialización de uno de los elementos.

```
<?php
$IdPers = ff_getParam('ff_param_view');
if (is_numeric($IdPers)){
    /*Recuperamos el parámetro correspondiente de la URL y devolvemos
    su valor como valor a mostrar en el campo*/
    $nombrePers = ff_getParam('ff_nombre_personal');
    return $nombrePers;
}
else
    return "No value available.";
?>
```

Tanto en el caso de que se acceda a la página de creación o modificación de personal por haber pulsado el botón de crear como en el caso de que se acceda para visualizar el detalle de un personal existente se debe ejecutar el código de inicialización del elemento correspondiente al campo del perfil al que está asociado el personal. En ambos casos el elemento debe mostrar los mismos valores (el listado de perfiles de usuario existentes en la aplicación) con la diferencia de que, en el caso de que se acceda por haber pulsado el botón de crear, se debe mostrar seleccionada por defecto la opción de “Seleccione perfil” mientras que en el caso de que se acceda para visualizar el detalle de un personal ya existente se debe mostrar seleccionada por defecto la opción asociada al perfil al que se encuentre vinculado el personal en cuestión. El código correspondiente a la inicialización del valor del campo “Perfil” se muestra en el siguiente cuadro de código fuente.

```
<?php
/*Recuperamos el listado de todos los perfiles de usuario existentes
en la base de datos*/
$perfilesExistentesCrear= ff_select(
    "select ".
    "Nombre, ID_Perfil ".
    "from perfiles ".
    "order by ID_Perfil "
);

/*Recuperamos el parámetro de la URL correspondiente al perfil que
nos va a permitir saber si se ha accedido para la creación de un
nuevo perfil (parámetro vacío) o para la visualización del detalle de
uno ya existente (parámetro con valor)*/
```

```

$perf = ff_getParam('ff_perfil');

if ($perf == ""){
    /*Si la variable correspondiente al parámetro está vacía significa
    que se ha pulsado la opción de crear un nuevo perfil y tenemos que
    mostrar todos los perfiles existentes en la aplicación mostrando
    seleccionada por defecto la opción "Seleccione perfil"*/
    $opcionesCrear = "1;Seleccione perfil;0";

    /*Recorremos el array obtenido de la consulta a la BBDD obteniendo el
    nombre de cada uno de los perfiles existentes e incluyendo dicho
    perfil como una opción de selección (no seleccionada) en la lista de
    valores*/
    foreach ($perfilesExistentesCrear as $perfilesECrear) {
        $nombrePerfilCrear = $perfilesECrear->Nombre;
        $idPerfilCrear = $perfilesECrear->ID_Perfil;
        $opcionesCrear .= "\n";
        $opcionesCrear .= "0;$nombrePerfilCrear;$idPerfilCrear";
    }

    //Devolvemos la lista de valores creada
    return $opcionesCrear;
}

else {
    /*Si la variable correspondiente al parámetro no está vacía
    significa que se ha accedido a visualizar el detalle de un personal
    existente y tenemos que mostrar todos los perfiles existentes en la
    aplicación mostrando seleccionada por defecto la opción
    correspondiente al perfil que tuviera el personal cuyo detalle se
    está visualizando*/
    $opcionesDetalle = "1;Seleccione perfil;0"; /*Recorremos el array obtenido
    de la consulta a la BBDD
    obteniendo el nombre de cada uno de los perfiles existentes
    e incluyendo dicho perfil como una posible opción de selección. En
    el caso de que el perfil recuperado coincida con el obtenido del
    parámetro de la URL dicho perfil se mostrará seleccionado por
    defecto.*/
    foreach ($perfilesExistentesCrear as $perfilesDetalle) {
        $nombrePerfilDetalle = $perfilesDetalle->Nombre;
        $idPerfilDetalle = $perfilesDetalle->ID_Perfil;
        if($nombrePerfilDetalle == $perf){
            $opcionesDetalle .= "\n";
            $opcionesDetalle .= "1;$nombrePerfilDetalle;$idPerfilDetalle";
        }
        else{
            $opcionesDetalle .= "\n";
            $opcionesDetalle .= "0;$nombrePerfilDetalle;$idPerfilDetalle";
        }
    }
    //Devolvemos la lista de valores creada
    return $opcionesDetalle;
}
?>

```


Como ya se ha comentado en varias ocasiones durante la presentación del código fuente implementado, el formulario de gestión de personal no cuenta con código a ejecutar después de cargar los elementos de la página por lo que, al igual que ocurría en el caso del código de la primera página del formulario, en este caso el siguiente código a ejecutar también será el asociado a los diferentes botones de acción existentes en la página de creación o visualización de un personal para la Asociación. Como puede observarse en la Figura 110 presentada al hablar de la interfaz de usuario de la aplicación, la página de creación o visualización de personal cuenta con cuatro botones asociados a las funcionalidades de guardar los cambios realizados sobre el personal, borrar un determinado personal, volver a la primera página del formulario y desconectar al usuario actual de la aplicación cuyo código será idéntico independientemente de si el acceso a la página ha sido para crear un nuevo personal o para modificar uno ya existente. Comenzamos presentando el código asociado al botón de guardar por ser el botón fundamental de la pantalla que, como se observa en el código siguiente, consiste únicamente en enviar los datos del formulario habiendo indicado anteriormente en la variable correspondiente que la acción ejecutada ha sido la de guardar los datos del personal.

```
function ff_btn_guardar_personal_action(element, action) {  
    switch (action) {  
        case 'click':  
            /*Debido a que cuando se produce el envío del formulario desde  
            un botón de acción siempre se ejecuta el mismo evento del  
            formulario es necesario poder identificar si el evento se ha  
            producido derivado de guardar cambios en un personal, de  
            borrar un personal, de realizar una búsqueda de personal o de  
            desconectar al usuario logado. Para ello, se emplea la  
            variable "Vble_oculta_accion_personal" que indica en cada  
            momento el botón pulsado para desencadenar la ejecución del  
            evento. En el caso de guardar cambios en un personal la  
            variable toma el valor uno*/  
            ff_getElementByName('Vble_oculta_accion_personal').value = 1;  
  
            //Enviamos los datos del formulario  
            ff_validate_submit();  
            break;  
        default:;  
    }  
}
```

Una vez que los datos del formulario han sido enviados al pulsar el botón de guardar se ejecuta el código correspondiente existente en el evento “Antes de enviar” del formulario que es donde realmente se realizan las comprobaciones y acciones necesarias para que la funcionalidad de guardar la creación de un nuevo personal o la modificación de uno ya existente tenga efecto. En líneas generales, el código asociado al botón de guardar que se muestra a continuación realiza el siguiente procedimiento: recupera los valores introducidos en cada uno de los campos de la pantalla y comprueba que todos se encuentren informados. A continuación realiza las comprobaciones pertinentes para garantizar tanto que el login dado al personal sea único en toda la aplicación como que solamente exista un personal asociado al perfil de superadministrador ya sea en la creación o en la modificación de personal. Finalmente, realiza la inserción del nuevo personal en la base de datos de la aplicación o la actualización del ya existente y redirecciona al usuario a la página de información

correspondiente o a la página de error oportuna en el caso de que alguna de las comprobaciones realizadas no haya sido superada.

```
/*Código a ejecutar al pulsar los botones de grabar, buscar, borrar o
desconectar*/
$this->execPieceByName('ff_InitLib');

/*Recuperamos el valor de la variable que indica si se ha pulsado el
botón de guardar, el botón de borrar, el botón de buscar o el botón
de desconectar*/
$vbleAccion = ff_getSubmit('Vble_oculta_accion_personal');

//Recuperamos los valores de personal enviados a través del
formulario
$valorNombrePersonal=ff_getSubmit('Valor_nombre_personal');
$valorLoginPersonal=ff_getSubmit('Valor_login');
$valorPassPersonal=ff_getSubmit('Valor_password');
//Encriptamos la clave recuperada del formulario
$passEncript = md5($valorPassPersonal);
$valorPerfilPersonal=ff_getSubmit('Valor_perfil');

if($vbleAccion == 1){
    //Estamos en el caso de que se ha pulsado el botón de guardar

    /*Comprobamos que se hayan introducido todos los campos necesarios
    para poder crear un nuevo personal*/
    if ($valorNombrePersonal == ''){
        /*Si no se ha introducido nombre para el personal mostramos el
        mensaje de error correspondiente tanto si se está creando un
        nuevo personal como si se está modificando uno ya existente*/
        $btnCrearModif = ff_getSubmit('Vble_oculta_crear_modificar');
        if($btnCrearModif == 1){
            //Crear nuevo personal
            ff_redirectForm('Crear_personal','&ff_page=8','get');
        }
        else{
            //Modificar personal ya existente
            ff_redirectForm('Crear_personal','&ff_page=12','get');
        }
    }
    else{
        /*En el caso de que se hubiera introducido nombre para el
        personal comprobamos si se ha introducido login*/
        if($valorLoginPersonal == ''){
            /*Si no se ha introducido login para el personal mostramos
            el mensaje de error correspondiente tanto si se está creando
            un nuevo personal como si se está modificando uno ya
            existente*/
            $btnCrearModif =
                ff_getSubmit('Vble_oculta_crear_modificar');
            if($btnCrearModif == 1){
                //Crear nuevo personal
                ff_redirectForm('Crear_personal','&ff_page=9','get');
            }
            else{
                //Modificar personal ya existente
                ff_redirectForm('Crear_personal','&ff_page=13','get');
            }
        }
    }
}
```

```
else{
    /*En el caso de que se hubiera introducido nombre y login
    para el personal comprobamos si se ha introducido password*/
    if($valorPassPersonal == ''){
        /*Si no se ha introducido password para el personal
        mostramos el mensaje de error correspondiente tanto si se
        está creando un nuevo personal como si se está
        modificando uno ya existente*/
        $btnCrearModif =
            ff_getSubmit('Vble_oculta_crear_modificar');
        if($btnCrearModif == 1){
            //Crear nuevo personal
            ff_redirectForm('Crear_personal','&ff_page=10','get');
        }
        else{
            //Modificar personal ya existente
            ff_redirectForm('Crear_personal','&ff_page=14','get');
        }
    }
    else{
        /*En el caso de que se hubiera introducido nombre, login
        y password para el personal comprobamos si se ha
        introducido perfil*/
        if($valorPerfilPersonal == 0){
            /*Si no se ha introducido perfil para el personal
            mostramos el mensaje de error correspondiente tanto si
            se está creando un nuevo personal como si se está
            modificando uno ya existente*/
            $btnCrearModif =
                ff_getSubmit('Vble_oculta_crear_modificar');
            if($btnCrearModif == 1){
                //Crear nuevo personal
                ff_redirectForm('Crear_personal','&ff_page=11','get'
                );
            }
            else{
                //Modificar personal ya existente
                ff_redirectForm('Crear_personal','&ff_page=15','get'
                );
            }
        }
        else{
            /*Se han introducido todos los campos necesarios para
            crear un nuevo personal. Comprobamos que no exista
            ningún otro personal con el login dado al nuevo
            personal a crear*/

            /*Recuperamos los logins de todo el personal existente
            en la BBDD*/
            $loginExistentes= ff_select(
                "select ".
                "Login ".
                "from personal "
            );

            /*Variable para indicar si el login introducido ya
            existe para algún personal ya existente*/
            $existeLogin = false;
```

```
/*Pasamos a minúsculas el valor del login introducido
para compararlo con el valor en minúsculas de todos y
cada uno de los logins del personal ya existente y así
evitar que se introduzca el mismo login dos veces, una
en mayúsculas y otra en minúsculas*/
$valorLoginPersonalMin =
    strtolower($valorLoginPersonal);

/*Recorremos el array obtenido de la consulta a la
BBDD obteniendo el login en minúscula de cada personal
existente y comparándolo con el introducido por el
usuario de manera que, si algún login coincide, al
salir del bucle la variable correspondiente tomará el
valor verdadero*/
foreach ($loginExistentes as $logins) {
    $loginCreado = $logins->Login;
    $loginCreadoMin = strtolower($loginCreado);
    if($loginCreadoMin == $valorLoginPersonalMin){
        $existeLogin = true;
        break;
    }
}

if($existeLogin == true){
    /*El login introducido por el usuario coincide con
    el de un personal ya existente. Comprobamos si se
    está creando un nuevo personal o modificando uno ya
    existente*/
    $btnCrearModif =
        ff_getSubmit('Vble_oculta_crear_modificar');

    if($btnCrearModif == 1){
        /*Se está creando un nuevo personal con un login
        igual al de un personal ya existente. Mostramos
        el error correspondiente*/
        ff_redirectForm('Crear_personal','&ff_page=3','get
        ');
    }
    else{
        /*Se está modificando un personal ya existente.
        Comprobamos si el login introducido coincide con
        el del personal que se está modificando o si
        coincide con el login de algún otro personal ya
        existente.*/

        /*Recuperamos el ID del personal que se está
        modificando y que ha sido enviado a través del
        formulario*/
        $idPersonalAct =
            ff_getSubmit('txt_pasar_ID_Personal');

        /*Recuperamos la información existente en la
        BBDD para el personal que se está modificando*/
        $loginBBDD= ff_select(
            "select ".
            "Login, PERFILES_ID_Perfil ".
            "from personal ".
            "where `ID_Personal` = '$idPersonalAct'"
        );
    }
}
```

```
foreach ($loginBBDD as $fila) {
    $loginPersBBDD = $fila->Login;
    $perfilPersBBDD = $fila->PERFILES_ID_Perfil;
}

/*Si el login introducido por el usuario coincide con el login del personal que se está modificando actualizamos la información del personal con la introducida por el usuario, sino mostramos el correspondiente mensaje de error*/
if ($valorLoginPersonal == $loginPersBBDD){
    /*Login introducido coincide con del personal modificado*/
    if($perfilPersBBDD == 1){
        /*Si el personal modificado es el superadministrador, su perfil no puede modificarse*/
        if ($valorPerfilPersonal != 1){
            /*Perfil del superadministrador modificado, mostramos mensaje de error*/
            ff_redirectForm('Crear_personal',
                '&ff_page=17','get');
        }
        else{
            /*Perfil del superadministrador no modificado, actualizamos los datos del personal*/
            ff_query(
                "update `personal` " .
                "set
                PERFILES_ID_Perfil='$valorPerfilPersonal',
                Nombre='$valorNombrePersonal',
                Login='$valorLoginPersonal',
                Password='$passEncrypt' " .
                "where ID_Personal =
                '$idPersonalAct'"
            );

            /*Mostramos mensaje de personal actualizado*/
            ff_redirectForm('Crear_personal',
                '&ff_page=19','get');
        }
    }
    else{
        /*Si el personal modificado no es el superadministrador, el perfil introducido no puede ser el de superadministrador*/
        if ($valorPerfilPersonal == 1){
            /*El perfil introducido es el de superadministrador, mostramos mensaje de error*/
            ff_redirectForm('Crear_personal',
                '&ff_page=18','get');
        }
        else{
            /*El perfil introducido no es el de superadministrador, actualizamos los datos del personal*/
        }
    }
}
```

```

        ff_query(
            "update `personal` " .
            "set
            PERFILES_ID_Perfil='$valorPerfilPersonal',
            Nombre='$valorNombrePersonal',
            Login='$valorLoginPersonal',
            Password='$passEncript' " .
            "where ID_Personal =
            '$idPersonalAct'"
        );

        /*Mostramos mensaje de personal
        actualizado*/
        ff_redirectForm('Crear_personal',
            '&ff_page=19','get');
    }
}
else{
    /*Si el login introducido por el usuario no
    coincide con el login del personal que se
    está modificando, pero coincide con el de
    otro personal ya existente mostramos el
    correspondiente mensaje de error*/
    ff_redirectForm('Crear_personal',
        '&ff_page=16','get');
}
}
}
else{
    /*El login introducido por el usuario no coincide
    con el de un personal ya existente. Comprobamos si
    se está creando un nuevo personal o modificando uno
    ya existente*/
    $btnCrearModif =
        ff_getSubmit('Vble_oculta_crear_modificar');

    if($btnCrearModif == 1){
        /*Se está creando un nuevo personal con un login
        distinto al de todos los existentes. Comprobamos
        si el perfil asignado es el de
        superadministrador*/
        if ($valorPerfilPersonal == 1){
            /*El perfil introducido es el de
            superadministrador. Comprobamos si ya existe
            un personal asociado a dicho perfil*/
            $perfilSuper= ff_select(
                "select ".
                "PERFILES_ID_Perfil ".
                "from personal "
            );

            /*Variable para indicar si ya existe un
            personal asociado al perfil de
            superadministrador*/
            $existeSuper = false;

```

```
/*Recorremos el array obtenido de la consulta
a la BBDD obteniendo el perfil de cada
personal existente y comprobando si es el de
superadministrador de manera que, si algún
perfil lo es, al salir del bucle la variable
correspondiente tomará el valor verdadero*/
foreach ($perfilSuper as $super) {
    $perfil = $super->PERFILES_ID_Perfil;
    if($perfil == 1){
        $existeSuper = true;
        break;
    }
}

if($existeSuper == true){
    /*Existe un usuario asociado al perfil de
superadministrador, mostramos el error ya
que no puede haber dos usuarios asociados
a dicho perfil*/
    ff_redirectForm('Crear_personal',
        '&ff_page=4','get');
}
}

/*El personal a crear no es superadministrador
o, si lo es, no existe otro usuario asociado a
dicho perfil. Creamos el nuevo personal*/
ff_query(
    "insert into `personal` " .
    "(" ID_Personal`,`PERFILES_ID_Perfil`,Nombre,
    Login,Password) " .
    "values
    (NULL,'$valorPerfilPersonal','$valorNombre
Personal','$valorLoginPersonal','$passEncri
pt') "
);

//Mostramos mensaje de personal creado
ff_redirectForm('Crear_personal',
    '&ff_page=5','get');
}
else{
    /*Se está modificando un personal existente con
un login distinto al de todos los existentes.
Comprobamos si el personal que se está
modificando es el de superadministrador*/

    /*Recuperamos el ID del personal que se está
modificando y que ha sido enviado a través del
formulario*/
    $idPersonalActN =
        ff_getSubmit('txt_pasar_ID_Personal');

    /*Recuperamos la información existente en la
BBDD para el personal que se está modificando*/
    $loginBBDDN= ff_select(
        "select " .
        "Login, PERFILES_ID_Perfil " .
        "from personal " .
        "where `ID_Personal` = '$idPersonalActN'"
    );
};
```

```
foreach ($loginBBDDN as $filaN) {
    $loginPersBBDDN = $filaN->Login;
    $perfilPersBBDDN =
        $filaN->PERFILES_ID_Perfil;
}

if($perfilPersBBDDN == 1){
    /*Si el personal que se está modificando es
    el de superadministrador, el perfil dado al
    modificar debe seguir siendo el de
    superadministrador*/
    if ($valorPerfilPersonal != 1){
        /*Perfil al modificar distinto de
        superadministrador, mostramos el error
        correspondiente*/
        ff_redirectForm('Crear_personal',
            '&ff_page=17','get');
    }
    else{
        /*Perfil al modificar sigue siendo el de
        superadministrador. Actualizamos el
        personal con la nueva información
        introducida*/
        ff_query(
            "update `personal` " .
            "set
                PERFILES_ID_Perfil='$valorPerfilPers
                onal',
                Nombre='$valorNombrePersonal',
                Login='$valorLoginPersonal',
                Password='$passEncript' " .
            "where ID_Personal = '$idPersonalActN'"
        );

        //Mostramos mensaje de personal modificado
        ff_redirectForm('Crear_personal',
            '&ff_page=19','get');
    }
}
else{
    /*Si el personal que se está modificando no
    es el de superadministrador, el perfil dado
    al modificar no puede ser el de
    superadministrador*/
    if ($valorPerfilPersonal == 1){
        /*Perfil al modificar es el de
        superadministrador, mostramos el error
        correspondiente*/
        ff_redirectForm('Crear_personal',
            '&ff_page=18','get');
    }
    else{
        /*Perfil al modificar no es el de
        superadministrador. Actualizamos el
        personal con la nueva información
        introducida*/
    }
}
```



```
ff_query(
    "update `personal` " .
    "set
        PERFILES_ID_Perfil='$valorPerfilPersonal',
        Nombre='$valorNombrePersonal',
        Login='$valorLoginPersonal',
        Password='$passEncript' " .
    "where ID_Personal = '$idPersonalActN'"
);

//Mostramos mensaje de personal modificado
ff_redirectForm('Crear_personal',
    '&ff_page=19','get');
}
}
}
}
}
}
}
}
```

La segunda de las acciones que es posible realizar desde la pantalla de creación o modificación de un personal es la de eliminar uno de los perfiles existentes. Llegados a este punto debería presentarse el código asociado al botón encargado de desencadenar la citada acción, pero debido a que el código es idéntico al presentado para el botón de guardar con la salvedad de que en esta ocasión la variable que indica la acción realizada tomará el valor dos (borrar) en lugar del valor uno (guardar) que tomaba en aquella ocasión se remite al código del botón guardar para su explicación centrándonos en este momento solamente en el código existente para la acción de borrado en el evento “Antes de enviar” del formulario. Dicho código consiste únicamente en comprobar si el personal que se desea borrar corresponde o no al personal vinculado al perfil de superadministrador y mostrar el error pertinente en el caso de que así sea o ejecutar el borrado del registro en la base de datos en el caso de que el personal a borrar no sea el superadministrador.

```
/*Código a ejecutar al pulsar los botones de grabar, buscar, borrar o
desconectar*/
$this->execPieceByName('ff_InitLib');

/*Recuperamos el valor de la variable que indica si se ha pulsado el
botón de guardar, el botón de borrar, el botón de buscar o el botón
de desconectar*/
$vbleAccion = ff_getSubmit('Vble_oculta_accion_personal');

//Recuperamos los valores de personal enviados a través del
formulario
$valorNombrePersonal=ff_getSubmit('Valor_nombre_personal');
$valorLoginPersonal=ff_getSubmit('Valor_login');
$valorPassPersonal=ff_getSubmit('Valor_password');
//Encriptamos la clave recuperada del formulario
$passEncrypt = md5($valorPassPersonal);
$valorPerfilPersonal=ff_getSubmit('Valor_perfil');

if($vbleAccion == 2){
    //Estamos en el caso de que se ha pulsado el botón de borrar

    /*Comprobamos si el perfil del personal a borrar es o no el de
    superadministrador, puesto que el superadministrador no puede
    borrarse*/
    if ($valorPerfilPersonal == 1){
        /*Usuario con perfil superadministrador, mostramos error
        indicando que no puede borrarse*/
        ff_redirectForm('Crear_personal','&ff_page=6','get');
    }
    else{
        //Ejecutamos la sentencia de borrado del personal seleccionado
        ff_query(
            "delete from `personal` " .
            "where `Login` = '$valorLoginPersonal'"
        );

        //Mostramos mensaje de personal borrado
        ff_redirectForm('Crear_personal','&ff_page=7','get');
    }
}
```

Continuando con la explicación del código asociado a cada uno de los botones de acción existentes en la página de creación o modificación de un personal se presenta a continuación el código correspondiente al botón de volver que, como su propio nombre indica, se encarga de mostrar nuevamente al usuario la página correspondiente al listado de personal existente en la Asociación anulando toda acción que se estuviera realizando en la página de creación o modificación.

```
function ff_btn_volver_personal_action(element, action) {
  switch (action) {
    case 'click':
      /*Reseteamos el valor de la variable que indica si se está
      creando un nuevo personal o modificando uno ya existente para
      garantizar que el nuevo flujo de funcionalidad que se comience
      funcione correctamente*/
      ff_getElementByName('Vble_oculta_crear_modificar').value = 0;

      /*Redireccionamos a la pagina de listado de personal en lugar
      de hacer un ff_previouspage() para asegurar que se borran los
      parámetros enviados a través de la URL y así garantizar que el
      nuevo flujo de creación o modificación que se comience funcione
      correctamente*/
      ff_redirect('http://localhost/joomla/index.php?option=com_facileforms&Itemid
      =6');
      break;
    default:;
  } // switch
}
```

De vuelta en la página de listado de personal existente en la Asociación se debe proceder a la explicación del código relativo al botón encargado de implementar la funcionalidad de búsqueda de personal. Al igual que ocurría en el caso de los botones de guardar o borrar un personal el código existente en el propio botón solamente se encarga de actualizar al valor correspondiente la variable encargada de determinar si se ha pulsado el botón de guardar, borrar, buscar o desconectar y de enviar los datos del formulario estando toda la funcionalidad propia del botón en cuestión implementada dentro del evento “Antes de enviar” del formulario. Por ello, a continuación se presenta únicamente el código existente en el ya citado evento. A grandes rasgos, el código existente en el mismo consiste en recuperar la información introducida en todos y cada uno de los campos de búsqueda existentes en la página de listado de personal y comprobar si alguno de ellos se encuentra informado. En el caso de que no se hubiera introducido ningún criterio de búsqueda el código del evento redirecciona a la propia página de listado de personal mostrando nuevamente todo el personal existente en la Asociación. Por el contrario, si se han introducido criterios de búsqueda, el código comprueba cuáles son dichos criterios y compone una consulta sobre el personal existente en la base de datos filtrando en función de los criterios de búsqueda existentes de manera que, al finalizar el código, se pueda conocer si existen o no registros de personal que satisfagan la búsqueda realizada. En el caso de que no los haya, la aplicación redireccionará a una nueva página de error indicando tal situación mientras que, en el caso de que si los haya, se redireccionará a la propia página de listado de personal enviando como parámetros a través de la URL los criterios de búsqueda introducidos de manera que sea el código de inicialización del elemento correspondiente al listado de personal, tal y como se vio al realizar su presentación, el encargado de mostrar al usuario el listado de personal que cumple la búsqueda introducida. Es decir, el código asociado al botón de buscar solamente se encarga de comprobar si existe o no personal que cumpla el criterio de búsqueda siendo el elemento de listado de personal el encargado de mostrar dicho personal a través de su código de inicialización.

```
/*Código a ejecutar al pulsar los botones de grabar, buscar, borrar o
desconectar*/
$this->execPieceByName('ff_InitLib');

/*Recuperamos el valor de la variable que indica si se ha pulsado el
botón de guardar, el botón de borrar, el botón de buscar o el botón
de desconectar*/
$vbLeAccion = ff_getSubmit('Vble_oculta_accion_personal');

if($vbLeAccion == 3){
    //Estamos en el caso de que se ha pulsado el botón de buscar

    /*Recuperamos los valores de búsqueda de personal enviados a
    través del formulario*/
    $persBuscarNombre = ff_getSubmit('Busq_personal_nombre');
    $persBuscarLogin = ff_getSubmit('Busq_personal_login');
    $persBuscarPerfil = ff_getSubmit('Busq_personal_perfil');

    /*Comprobamos si se ha introducido información para buscar en
    alguno de los campos de búsqueda*/
    if($persBuscarNombre == '' && $persBuscarLogin == '' &&
    $persBuscarPerfil == 0){
        /*Si no se ha introducido ningún criterio de búsqueda
        redireccionamos a la página actual de listado de personal
        mostrando todos los registros de personal existentes en la
        BBDD*/
        ff_redirectSelf('&ff_page=1', 'get');
    }

    else{
        /*Se han introducido criterios de búsqueda. Debemos comprobar
        qué campos de búsqueda se han introducido para componer en cada
        caso la consulta a la BBDD adecuada filtrando en función de los
        criterios de búsqueda introducidos*/

        /*Variable para indicar si existe algún personal en la BBDD que
        cumpla los criterios de búsqueda introducidos por el usuario*/
        $existeNombrePers = 0;

        if($persBuscarNombre != ''){
            //Criterio de búsqueda por nombre de personal introducido
            if($persBuscarLogin != ''){
                //Criterio de búsqueda por login de personal introducido
                if($persBuscarPerfil != 0){
                    /*Criterio de búsqueda por perfil de personal
                    introducido*/

                    /*Ejecutamos consulta a la BBDD filtrando por nombre,
                    login y perfil de personal. Para que las búsquedas no
                    sean exactas introducimos el carácter comodín "%"
                    antes y después de la variable del filtro de
                    búsqueda*/
                    $registroPers = ff_select(
                        "select ".
                        "Nombre, Login, PERFILES_ID_Perfil ".
                        "from personal ".
                        "where Nombre like '%" . $persBuscarNombre . "%' and
                        Login like '%" . $persBuscarLogin . "%' and
                        PERFILES_ID_Perfil like '$persBuscarPerfil' "
                    );
                }
            }
        }
    }
}
```

```
/*Variable que indica si existe algún personal cuyo
perfil contenga el introducido en la búsqueda*/
$estaP = false;

/*Recorremos el array obtenido de la consulta a la
BBDD obteniendo el nombre, login y perfil de cada
personal existente y comprobando si contiene el
criterio de búsqueda correspondiente de manera que, si
algún personal coincide con todos los criterios de
búsqueda introducidos, al salir del bucle la variable
correspondiente tomará el valor verdadero*/
foreach ($registroPers as $regPers) {
    $nombrePersAlmacenado = $regPers->Nombre;
    $loginPersAlmacenado = $regPers->Login;
    $perfilPersAlmacenado =
        $regPers->PERFILES_ID_Perfil;

    /*La función stripos devuelve la primera posición
    del string pasado como segundo argumento en la que
    encuentra el string pasado como primer argumento,
    realizando una búsqueda no sensitiva a mayúsculas y
    minúsculas*/
    $estaN = stripos($nombrePersAlmacenado,
        $persBuscarNombre);
    $estaL = stripos($loginPersAlmacenado,
        $persBuscarLogin);

    if($perfilPersAlmacenado == $persBuscarPerfil){
        $estaP = true;
    }
    if($estaN != false && $estaL != false && $estaP
    != false){
        $existeNombrePers = 1;
        break;
    }
}
}
else{
    /*Criterio de búsqueda por perfil de personal no
    introducido*/

    /*Ejecutamos consulta a la BBDD filtrando por nombre y
    login de personal. Para que las búsquedas no sean
    exactas introducimos el carácter comodín "%" antes y
    después de la variable del filtro de búsqueda*/
    $registroPers = ff_select(
        "select ".
        "Nombre, Login, PERFILES_ID_Perfil ".
        "from personal ".
        "where Nombre like '%" . $persBuscarNombre . "%' and
        Login like '%" . $persBuscarLogin . "%' "
    );
    /*Recorremos el array obtenido de la consulta a la
    BBDD obteniendo el nombre y login de cada personal
    existente y comprobando si contiene el criterio de
    búsqueda correspondiente de manera que, si algún
    personal coincide con todos los criterios de búsqueda
    introducidos, al salir del bucle la variable
    correspondiente tomará el valor verdadero*/
}
```

```
foreach ($registroPers as $regPers) {
    $nombrePersAlmacenado = $regPers->Nombre;
    $loginPersAlmacenado = $regPers->Login;

    /*La función stripos devuelve la primera posición
    del string pasado como segundo argumento en la que
    encuentra el string pasado como primer argumento,
    realizando una búsqueda no sensitiva a mayúsculas y
    minúsculas*/
    $estaN = stripos($nombrePersAlmacenado,
        $persBuscarNombre);
    $estaL = stripos($loginPersAlmacenado,
        $persBuscarLogin);

    if($estaN !== false && $estaL !== false){
        $existeNombrePers = 1;
        break;
    }
}
}
}
else{
    /*Criterio de búsqueda por login de personal no
    introducido*/
    if($persBuscarPerfil != 0){
        /*Criterio de búsqueda por perfil de personal
        introducido*/

        /*Ejecutamos consulta a la BBDD filtrando por nombre y
        perfil de personal. Para que las búsquedas no sean
        exactas introducimos el carácter comodín "%" antes y
        después de la variable del filtro de búsqueda*/
        $registroPers = ff_select(
            "select ".
            "Nombre, Login, PERFILES_ID_Perfil ".
            "from personal ".
            "where Nombre like '%" . $persBuscarNombre . "%' and
            PERFILES_ID_Perfil like '$persBuscarPerfil' "
        );

        /*Variable que indica si existe algún personal cuyo
        perfil contenga el introducido en la búsqueda*/
        $estaP = false;

        /*Recorremos el array obtenido de la consulta a la
        BBDD obteniendo el nombre y perfil de cada personal
        existente y comprobando si contiene el criterio de
        búsqueda correspondiente de manera que, si algún
        personal coincide con todos los criterios de búsqueda
        introducidos, al salir del bucle la variable
        correspondiente tomará el valor verdadero*/
        foreach ($registroPers as $regPers) {
            $nombrePersAlmacenado = $regPers->Nombre;
            $perfilPersAlmacenado =
                $regPers->PERFILES_ID_Perfil;

            /*La función stripos devuelve la primera posición
            del string pasado como segundo argumento en la que
            encuentra el string pasado como primer argumento,
            realizando una búsqueda no sensitiva a mayúsculas y
            minúsculas*/
```

```

        $ستاN = stripos($nombrePersAlmacenado,
            $persBuscarNombre);

        if($perfilPersAlmacenado == $persBuscarPerfil){
            $ستاP = true;
        }
        if($ستاN != false && $ستاP != false){
            $existeNombrePers = 1;
            break;
        }
    }
}
else{
    /*Criterio de búsqueda por perfil de personal no
    introducido*/

    /*Ejecutamos consulta a la BBDD filtrando por nombre
    de personal. Para que las búsquedas no sean exactas
    introducimos el carácter comodín "%" antes y después
    de la variable del filtro de búsqueda*/
    $registroPers = ff_select(
        "select ".
        "Nombre, Login, PERFILES_ID_Perfil ".
        "from personal ".
        "where Nombre like '%" . $persBuscarNombre . "%' "
    );

    /*Recorremos el array obtenido de la consulta a la
    BBDD obteniendo el nombre de cada personal existente y
    comprobando si contiene el criterio de búsqueda
    correspondiente de manera que, si algún personal
    coincide con todos los criterios de búsqueda
    introducidos, al salir del bucle la variable
    correspondiente tomará el valor verdadero*/
    foreach ($registroPers as $regPers) {
        $nombrePersAlmacenado = $regPers->Nombre;
        $ستاN = stripos($nombrePersAlmacenado,
            $persBuscarNombre);

        if($ستاN != false){
            $existeNombrePers = 1;
            break;
        }
    }
}
}
else{
    /*Criterio de búsqueda por nombre de personal no
    introducido*/
    if($persBuscarLogin != ''){
        /*Criterio de búsqueda por login de personal introducido
        if($persBuscarPerfil != 0){
            /*Criterio de búsqueda por perfil de personal
            introducido*/

            /*Ejecutamos consulta a la BBDD filtrando por login y
            perfil de personal. Para que las búsquedas no sean
            exactas introducimos el carácter comodín "%" antes y
            después de la variable del filtro de búsqueda*/

```

```
$registroPers = ff_select(
    "select ".
    "Nombre, Login, PERFILES_ID_Perfil ".
    "from personal ".
    "where Login like '%" . $persBuscarLogin . "%' and
    PERFILES_ID_Perfil like '$persBuscarPerfil' "
);

/*Variable que indica si existe algún personal cuyo
perfil contenga el introducido en la búsqueda*/
$estaP = false;

/*Recorremos el array obtenido de la consulta a la
BBDD obteniendo el login y perfil de cada personal
existente y comprobando si contiene el criterio de
búsqueda correspondiente de manera que, si algún
personal coincide con todos los criterios de búsqueda
introducidos, al salir del bucle la variable
correspondiente tomará el valor verdadero*/
foreach ($registroPers as $regPers) {
    $loginPersAlmacenado = $regPers->Login;
    $perfilPersAlmacenado =
        $regPers->PERFILES_ID_Perfil;

    /*La función stripos devuelve la primera posición
    del string pasado como segundo argumente en la que
    encuentra el string pasado como primer argumento,
    realizando una búsqueda no sensitiva a mayúsculas y
    minúsculas*/
    $estaL = stripos($loginPersAlmacenado,
        $persBuscarLogin);

    if($perfilPersAlmacenado == $persBuscarPerfil){
        $estaP = true;
    }
    if($estaL !== false && $estaP !== false){
        $existeNombrePers = 1;
        break;
    }
}
}
else{
    /*Criterio de búsqueda por perfil de personal no
    introducido*/

    /*Ejecutamos consulta a la BBDD filtrando por login de
    personal. Para que las búsquedas no sean exactas
    introducimos el carácter comodín "%" antes y después
    de la variable del filtro de búsqueda*/
    $registroPers = ff_select(
        "select ".
        "Nombre, Login, PERFILES_ID_Perfil ".
        "from personal ".
        "where Login like '%" . $persBuscarLogin . "%' "
    );
};
```



```
/*Recorremos el array obtenido de la consulta a la
BBDD obteniendo el login de cada personal existente y
comprobando si contiene el criterio de búsqueda
correspondiente de manera que, si algún personal
coincide con todos los criterios de búsqueda
introducidos, al salir del bucle la variable
correspondiente tomará el valor verdadero*/
foreach ($registroPers as $regPers) {
    $loginPersAlmacenado = $regPers->Login;

    /*La función strpos devuelve la primera posición
    del string pasado como segundo argumento en la que
    encuentra el string pasado como primer argumento,
    realizando una búsqueda no sensitiva a mayúsculas y
    minúsculas*/
    $estaL = strpos($loginPersAlmacenado,
        $persBuscarLogin);

    if($estaL !== false){
        $existeNombrePers = 1;
        break;
    }
}
}
else{
    /*Criterio de búsqueda por login de personal no
    introducido*/
    if($persBuscarPerfil != 0){
        /*Criterio de búsqueda por perfil de personal
        introducido*/

        /*Ejecutamos consulta a la BBDD filtrando por perfil
        de personal. Para que las búsquedas no sean exactas
        introducimos el carácter comodín "%" antes y después
        de la variable del filtro de búsqueda*/
        $registroPers = ff_select(
            "select ".
            "Nombre, Login, PERFILES_ID_Perfil ".
            "from personal ".
            "where PERFILES_ID_Perfil like '$persBuscarPerfil'"
        );

        /*Variable que indica si existe algún personal cuyo
        perfil contenga el introducido en la búsqueda*/
        $estaP = false;

        /*Recorremos el array obtenido de la consulta a la
        BBDD obteniendo el perfil de cada personal existente y
        comprobando si coincide con el criterio de búsqueda
        correspondiente de manera que, si algún personal
        coincide con todos los criterios de búsqueda
        introducidos, al salir del bucle la variable
        correspondiente tomará el valor verdadero*/
    }
}
```

```

        foreach ($registroPers as $regPers) {
            $perfilPersAlmacenado =
                $regPers->PERFILES_ID_Perfil;

            if($perfilPersAlmacenado == $persBuscarPerfil){
                $existeNombrePers = 1;
                break;
            }
        }
    }
}

/*Comprobamos si existe personal en la BBDD que cumpla los
criterios de búsqueda introducidos por el usuario*/
if($existeNombrePers == 0){
    /*No existe personal en la BBDD que cumpla los criterios de
    búsqueda introducidos por el usuario, mostramos el
    correspondiente mensaje de error*/
    ff_redirectForm('Crear_personal','&ff_page=20','get');
}
else{
    /*Existe personal en la BBDD que cumple los criterios de
    búsqueda introducidos por el usuario. Redireccionamos a la
    página actual de listado de personal pasando como parámetros
    por la URL el nombre, login y perfil introducidos como
    criterios de búsqueda de manera que al cargar de nuevo la
    página se ejecute aplicando el filtro correspondiente y solo
    muestre los registros de personal que cumplen la búsqueda
    introducida*/
    ff_redirectSelf("&ff_nombre_personal=$persBuscarNombre
        &ff_login=$persBuscarLogin&ff_perfil=$persBuscarPerfil
        ","get");
}
}
}

```

Para finalizar con la presentación del código vinculado a las diferentes acciones que es posible realizar sobre los elementos del formulario y, debido a que no existe código a ejecutar en los eventos de “Script enviado” y “Después de enviar” del formulario, también para finalizar con la presentación del código correspondiente a la implementación de la funcionalidad de gestión de personal de la Asociación se explica a continuación el código encargado de realizar la funcionalidad de desconexión del usuario logado en la aplicación. De la misma manera que en el caso de los botones de crear, borrar y buscar ya presentados, en el caso de la función de desconexión que ahora nos ocupa el código del botón propiamente dicho sólo se encarga de establecer el valor apropiado en la variable que indica el botón pulsado y de enviar los datos del formulario, siendo el evento “Antes de enviar” el encargado de realizar la funcionalidad de desconexión propiamente dicha, tal y como se observa en el código siguiente.

```
/*Código a ejecutar al pulsar los botones de grabar, buscar, borrar o
desconectar*/
$this->execPieceByName('ff_InitLib');

/*Recuperamos el valor de la variable que indica si se ha pulsado el
botón de guardar, el botón de borrar, el botón de buscar o el botón
de desconectar*/
$vbLeAccion = ff_getSubmit('Vble_oculta_accion_personal');

if($vbLeAccion == 4){
    //Estamos en el caso de que se ha pulsado el botón de desconectar

    /*Para poder gestionar correctamente la aparición o no del menú en
función de si el usuario se encuentra logado o no necesitamos
comparar la información introducida por el usuario para logarse
con la existente en la BBDD para cada usuario. Desde el menú no es
posible acceder a la información introducida por el usuario para
logarse, por lo que es necesario almacenar esa información
mientras el usuario esté logado en una tabla temporal a la que
accederá el menú para realizar la comprobación y mostrar o no el
menú. Cuando el usuario se desconecta borramos la información de
la tabla temporal para evitar que el usuario que se intente logar
después visualice el menú aunque introduzca mal los datos de
acceso a la aplicación*/
    $accesoB= ff_select(
        "delete ".
        "from accesotmp "
    );

    //Redireccionamos a la página de login
    ff_redirectForm('Formulario_login','&ff_page=1','get');
}
```

Con la explicación del código asociado a la funcionalidad de desconexión que se acaba de realizar se da por finalizada la exposición del procedimiento de programación utilizado por el componente FacileForms de Joomla y seguida en el caso concreto de la implementación de la opción de gestión de personal expuesta. Llegados a este punto se debe mencionar que dicho procedimiento presentado para el caso de la gestión de personal es el mismo que se ha seguido en la implementación de la funcionalidad de gestión de perfiles también implementada dentro del pequeño prototipo que debía alcanzar el proyecto y que será el mismo que deberá seguirse en la realización del resto de funcionalidades con que debe contar la aplicación de la Asociación Navarra de Síndrome de Down. De la misma manera, debe indicarse que la realización del menú mostrado en la aplicación difiere del procedimiento aquí explicado puesto que su implementación no ha sido realizada con el componte FacileFoms sino empleando el módulo MetaMod [MetaMod] de Joomla. Por este motivo, la implementación de dicho menú se explica de manera independiente en el siguiente apartado.

4.3.- Implementación “Menú”


El principal motivo por el que la implementación del menú requiere de un apartado específico es debido a que era necesario que dicho menú con las opciones disponibles en la aplicación sólo estuviera visible una vez que un usuario registrado en la herramienta se hubiera logado en la misma y esta funcionalidad no era soportada de manera nativa por la gestión de menús que ofrece Joomla. Para solucionar este inconveniente es necesario recurrir al módulo MetaMod que se trata de un módulo adicional de Joomla que, conjuntamente con el módulo estándar de gestión de menús, permite implementar la funcionalidad deseada para el menú, es decir, permite que el menú sólo muestre las opciones disponibles una vez que el usuario se haya logado en la aplicación.

Después de haber introducido la problemática existente con la funcionalidad del menú de la aplicación se va a proceder a continuación a presentar la implementación llevada a cabo para disponer del citado menú, implementación realizada en su mayor parte a través del módulo de gestión de menús estándar de Joomla y sin necesidad de introducir código adicional. El punto de partida para la implementación del menú consiste en la definición de las distintas opciones con que contará la aplicación a desarrollar. Para ello, es necesario crear una nueva instancia del módulo de gestión de menús existente dentro del panel de administración de menús de Joomla. Una vez que dicha instancia ha sido creada aparece como una opción más dentro de la opción “Menús” de la barra de herramientas de Joomla. Accediendo a dicha opción recién creada se visualiza una pantalla similar a la siguiente donde el usuario podrá ir introduciendo todas las opciones de menú que desee.

#	Artículo del menú	Publicado	Reordenar	Orden	Accesos	Itemid	Tipos	CID
1	Personas			3	Public	13	Componente	19
2	Programas			9	Public	14	Componente	19
3	Actividades			10	Public	15	Componente	19
4	Usuarios-Actividades			11	Public	16	Componente	19
5	Entidades			12	Public	17	Componente	19
6	Subvenciones			13	Public	18	Componente	19
7	Proveedores			14	Public	19	Componente	19
8	Facturas			15	Public	20	Componente	19
9	Salarios			16	Public	21	Componente	19
10	Gastos extraordinarios			17	Public	22	Componente	19
11	Actividades extraordinarias			18	Public	23	Componente	19
12	Perfiles			19	Public	3	Componente	19
13	Personal			20	Public	6	Componente	19
14	Cambio password			21	Public	24	Componente	19

Figura 119. Creación menú aplicación

La creación de las distintas opciones del menú mostradas en la figura anterior se realiza a través del botón “Nuevo” existente para ello en la barra de herramientas del administrador de menús de Joomla. Pulsando en dicho botón el usuario dispone de una pantalla como la mostrada en la Figura 120 donde poder elegir el tipo de elemento que se introducirá en el menú.



Nuevo Artículo de Menú

* Algunos tipos de menú aparecen en más de un grupo, pero continúan siendo el mismo tipo de menú

Contenido

Bloque - Categoría de contenido	<input type="radio"/>
Bloque - Categoría de contenidos archivados	<input type="radio"/>
Bloque - Contenido de sección archivada	<input type="radio"/>
Bloque - Sección de contenido	<input type="radio"/>
Enlace - Artículo de contenido	<input type="radio"/>
Enlace - Contenido estático	<input type="radio"/>
Enviar - Contenido	<input type="radio"/>
Tabla - Categoría de contenidos	<input type="radio"/>
Tabla - Sección de contenido	<input type="radio"/>

Componentes

Componente	<input type="radio"/>
Enlace - Artículo de contacto	<input type="radio"/>
Enlace - Componente	<input type="radio"/>
Enlace - Noticia externa	<input type="radio"/>
Tabla - Categoría de contactos	<input type="radio"/>
Tabla - Categoría de enlaces web	<input type="radio"/>
Tabla - Categoría de noticias externas	<input type="radio"/>

Miscelánea

Separador	<input type="radio"/>
Wrapper	<input type="radio"/>

Enviar

Enviar - Contenido	<input type="radio"/>
--------------------	-----------------------

Enlaces

Enlace - Artículo de contacto	<input type="radio"/>
Enlace - Artículo de contenido	<input type="radio"/>
Enlace - Componente	<input type="radio"/>
Enlace - Contenido estático	<input type="radio"/>
Enlace - Noticia externa	<input type="radio"/>
Enlace - Url	<input type="radio"/>

Figura 120. Selección artículos de menú

En el caso del menú para la aplicación de la Asociación Navarra de Síndrome de Down que nos ocupa, todas las opciones de menú creadas son de tipo “Componente” por ser el elemento que permite vincular una determinada opción de menú con los distintos formularios creados a través del componente FacileForms. Al seleccionar dicho tipo de elemento se presenta al usuario una nueva ventana de configuración similar a la siguiente donde poder parametrizar diferentes aspectos de la opción del menú que se está creando.



Editor Artículo del menú :: Componente [FacileForms]

Detalles

Name:

Componente: FacileForms

URL:

Artículo padre:

Top

Personas
 Programas
 Actividades
 Usuarios-Actividades
 Entidades
 Subvenciones
 Proveedores
 Facturas
 Salarios

Orden:

Nivel de acceso:

Public

Registered

Special

Publicar:
 ☐ No
 ☒ Sí

Parámetros

FacileForms - Parameter settings

Form name

Page number

Run in iframe
 ☒ No
 ☐ Yes

Border
 ☐ No
 ☒ Yes

Horizontal align

Left position (pixels)

Top position (pixels)

Private parameters

Public parameters

CSS class suffix

Figura 121. Configuración artículos de menú

En la imagen anterior se observa que la configuración de la opción del menú a crear pasa por establecer, entre otros detalles, el nombre que presentará la opción cuando se muestre en el menú, determinar si se tratará de una opción principal o si será una subsección dentro de otra opción existente en el menú, indicar la posición que ocupará la opción dentro del menú, el tipo de acceso que el usuario tendrá sobre la misma y si estará o no publicada. Todo ello relacionado con el aspecto visual de la opción del menú a crear. En lo que a aspectos funcionales se refiere, la configuración de la opción del menú permite indicar el formulario, y más concretamente el número de página del formulario, que se mostrará en la parte central de la ventana de la aplicación en el momento en el que el usuario pulse sobre la opción del menú correspondiente. Es importante indicar en este momento que el nombre del formulario aquí indicado debe coincidir con el creado para dar respuesta a una funcionalidad determinada a través del componente FacileForms de Joomla, tal y como se explicó en el apartado anterior del capítulo actual.

Realizando el proceso que se acaba de describir para cada una de las opciones que se desea disponga la aplicación, se obtiene el menú que se muestra en la figura siguiente.



Figura 122. Menú aplicación

La creación del menú tal y como se acaba de presentar cuenta con el inconveniente de que dicho menú se muestra en todo momento de la misma manera en que aparece en la Figura 122, lo que significa que aquellos usuarios que no se encuentren logados en la herramienta podrán disponer de la información que indica las distintas opciones con que cuenta la aplicación. Para solventar esta situación se instaló en Joomla el módulo MetaMod según el proceso explicado en la referencia [Inst.MetaMod]. Se trata de un “metamódulo” para Joomla que incluye dentro de él otros módulos creados en Joomla que muestra en función de determinadas reglas de visionado o parámetros definidos por el usuario. En el caso que nos ocupa el módulo

MetaMod debe encargarse de mostrar el módulo anteriormente creado únicamente en los casos en los que un usuario haya accedido correctamente a la herramienta, mostrando mientras el usuario no se encuentre logado un menú vacío. Para ello, en primer lugar, es necesario proceder a la creación de una nueva instancia del menú estándar de Joomla que no contará con ninguna opción disponible y que será el menú que la aplicación mostrará mientras no haya un usuario logado (la creación de dicho menú se realiza según el procedimiento presentado para la creación del menú que cuenta con las opciones de la aplicación). Una vez que se dispone de los dos menús a mostrar y del módulo MetaMod instalado, se deberá acceder a cada uno de los módulos creados en el administrador de módulos de Joomla al crear los dos menús de la aplicación. Al acceder a ellos, se muestra la página de configuración de los mismos (Figura 123) donde deberá indicarse que el menú no debe estar asociado a ninguna de las páginas existentes en la aplicación. De esta manera garantizamos que los menús creados no se muestren por defecto y su aparición se gestione únicamente a través del módulo MetaMod.

Sitio Módulo del: Editar [Menu_prueba]

Detalles

Título:

Mostrar título: ☒ No ☐ Sí

Posición:

Orden del Módulo:

Nivel de acceso:

Publicado: ☐ No ☒ Sí

ID: 41

Descripción: Muestra el menú principal.

Parámetros

Estilo CSS para el menú:

Estilo CSS para el módulo:

Nombre del menú:

Estilo del menú:

Páginas / Artículos

Artículos del menú:

- Todo
- Ninguno
- Sin asignar
- mainmenu | Asociación Síndrome de Down
- MenuAplicacion | Personas
- MenuAplicacion | Programas
- MenuAplicacion | Actividades
- MenuAplicacion | Usuarios-Actividades
- MenuAplicacion | Entidades
- MenuAplicacion | Subvenciones
- MenuAplicacion | Proveedores
- MenuAplicacion | Facturas
- MenuAplicacion | Salarios
- MenuAplicacion | Gastos extraordinarios
- MenuAplicacion | Actividades extraordinarias
- MenuAplicacion | Perfiles
- MenuAplicacion | Personal
- MenuAplicacion | Cambio password

Figura 123. Configuración módulo menú

A continuación, y también desde el panel de administrador de módulos de Joomla, debe accederse al enlace correspondiente al módulo MetaMod que se habrá creado después de haber instalado dicho módulo en Joomla. Es a través de este enlace donde se realizará la determinación de en qué casos deberá mostrarse el menú vacío o el menú que muestra las opciones existentes en la aplicación. Para ello, en primer lugar, se debe indicar que el módulo MetaMod debe asignarse a todas las páginas existentes en la aplicación y, una vez hecho esto, en la sección PHP existente en su página de configuración deberá programarse el código que se desee que determine el criterio de visualización de un menú u otro. En el caso concreto que nos ocupa, la condición que determina que se muestre un menú u otro es el hecho de que el usuario se encuentre o no logado en la aplicación por lo que el código PHP que habrá que asignar al módulo MetaMod deberá recuperar el login y password de todo el personal existente en la aplicación y compararlo con el login y password introducido por el usuario de manera

que si éste último login y password coincide con alguno de los existentes en la aplicación se muestre el menú con las opciones disponibles mientras que si no coincide se muestre el menú vacío. El detalle del código encargado de realizar dicha comprobación se muestra a continuación.

```
/*Para determinar el menú a mostrar se debe comprobar si el login y
password introducido por el usuario coincide con alguno de los
existentes para el personal registrado en la aplicación*/

/*Recuperamos todos los registros correspondientes al personal
existentes en la base de datos*/
$personal=mysql_query(
    "select PERFILES_ID_Perfil, Nombre, Login, Password
    from personal;"
);

/*Variables para almacenar el login y password introducidos por el
usuario al intentar logarse en la aplicación*/
$loginInt = "";
$passInt = "";

/*Recuperamos el login y password introducidos por el usuario al
intentar acceder a la aplicación. Dicho login y password habrá sido
almacenado en una tabla temporal en el momento en el que el usuario
realiza el login para poder recuperarlos en este momento*/
$actual=mysql_query(
    "select LoginAcceso, PassAcceso
    from accesotmp;"
);

/*La tabla de login y password temporal siempre va a contar con un
único registro por lo que almacenos su información en las variables
creadas para ello para así poder comparar dichos valores con los
existentes en la tabla de personal*/
while($row = mysql_fetch_array($actual)) {
    $loginInt = $row['LoginAcceso'];
    $passInt = $row['PassAcceso'];
}

/*Variable para indicar si el login y password introducido por el
usuario al logarse coincide o no con los datos de acceso de algún
personal de la aplicación*/
$coincide = false;

/*Recorremos el array obtenido de la consulta a la BBDD obteniendo el
login y password de cada personal existente y comparándolo con el
login y password introducido por el usuario al logarse de manera que,
si este login y password coincide con alguno de los registrador, al
salir del bucle la variable correspondiente tomará el valor verdadero
```



```
while($rowP = mysql_fetch_array($personal)) {  
    $loginPersonal = $rowP['Login'];  
    $passPersonal = $rowP['Password'];  
    if($loginPersonal == $loginInt){  
        if($passPersonal == $passInt){  
            $coincide = true;  
            break;  
        }  
    }  
}  
  
/*Si el login y password introducido por el usuario al logarse  
coincide con alguno de los existentes para el personal mostramos el  
menú con las opciones de la aplicación, sino mostramos el menú  
vacío*/  
if($coincide == true){  
    return 34;  
}  
else{  
    return 41;  
}
```

Así, conseguimos que la gestión del menú se realice de acuerdo a los requisitos exigidos por la Asociación Navarra de Síndrome de Down. Esto es, con la ayuda del módulo MetaMod, y siguiendo las indicaciones establecidas en la referencia [MundoJoomla], se consigue que el menú con las opciones disponibles en la aplicación sólo sea visible para los usuarios logados en la herramienta. Así mismo, y esta vez debido al módulo estándar de gestión de menús de Joomla, se ha conseguido la vinculación de cada opción del menú con el formulario correspondiente creado en FacileForms, logrando así disponer de una herramienta dinámica para el usuario.

Capítulo 5. Conclusiones y líneas futuras

En este último capítulo de la memoria descriptiva del proyecto elaborado, se van a presentar todas aquellas conclusiones que se han obtenido de la realización del mismo así como los aspectos que, por diversos motivos, no se han llevado a cabo y sería necesario desarrollar en futuras versiones de la herramienta para una correcta satisfacción de los requisitos establecidos en el segundo capítulo de esta misma memoria.

Como primera conclusión a extraer del trabajo realizado en este proyecto se debe indicar que el resultado del mismo es satisfactorio puesto que se ha logrado analizar correctamente la problemática existente en la Asociación y diseñar una herramienta que, según el *feedback* dado por el usuario, permite dar respuesta a la demanda de una correcta gestión informática del trabajo llevado a cabo dentro de la Asociación. Es decir, se considera que la herramienta desarrollada, y que será implementada en un futuro, puede ser de gran utilidad en la labor diaria del colectivo de trabajadores de la Asociación Navarra de Síndrome de Down facilitándoles en gran medida el trabajo de gestión administrativa que conlleva una entidad de esta envergadura.

Centrándonos ahora en las conclusiones extraídas del procedimiento y herramientas de trabajo empleadas se debe decir que inicialmente, al comienzo del proyecto, se pensaba que la parte costosa y complicada del mismo iba a ser la dedicada al análisis y diseño de la solución mientras que se consideraba que la implementación del prototipo con la herramienta Joomla iba a ser algo más fácil y menos costoso de realizar. Sin embargo, al final del proyecto, podemos decir que ha sido todo lo contrario. Es decir, la parte de análisis y diseño no fue del todo complicada puesto que el usuario conocía perfectamente sus carencias y el objetivo que quería alcanzar con la realización de este trabajo y supo expresarlo claramente en las reuniones de captura de requisitos mantenidas al comienzo del proyecto lo que repercutió en que no fuera necesario realizar grandes modificaciones en el diseño de la solución inicial. Por el contrario, la implementación del pequeño prototipo de la herramienta ha requerido mucho más tiempo del que inicialmente se preveía y es que Joomla es un Gestor de Contenidos muy potente y recomendable, pero en el momento en que es necesario emplearlo para

implementar una aplicación con funcionalidad propia, interacción entre páginas y comunicación con base de datos no es tan intuitivo y sencillo de manejar como en un principio pudiera parecer. Así, en primer lugar, fue complicado comprender el funcionamiento interno del componente empleado para la creación de formularios, averiguar el flujo de ejecución que se seguía y dónde debía situarse el código para obtener los resultados deseados. Por otra parte, no ha sido fácil encontrar información sobre los diferentes elementos que podían emplearse como parte de un formulario costando especialmente la implementación de los elementos de tipo listado y lista de valores. Igualmente complicado ha sido el paso de parámetros entre páginas para la implementación de la funcionalidad de búsqueda y la implementación del menú de manera que sólo se mostrara en el momento en que el usuario se logara en la aplicación. En general, la implementación del prototipo ha requerido la realización de una gran labor de investigación para conseguir sacar adelante cualquiera de las funcionalidades a implementar lo que ha provocado que su construcción haya costado más de lo previsto.

Para finalizar con la parte del capítulo dedicada a las conclusiones, se debe indicar que la realización del presente proyecto me ha sido de gran utilidad puesto que, aparte de permitirme conocer la labor llevada a cabo por la Asociación Navarra de Síndrome de Down, me ha dado la oportunidad de repasar y poner en práctica muchos de los conocimientos teóricos, métodos y herramientas adquiridos durante el estudio de la carrera, fundamentalmente relacionados con la programación en PHP y con la ingeniería del software en temas relativos al análisis y diseño de aplicaciones. Además, me ha dado a conocer la realidad laboral puesto que me ha permitido descubrir la manera en que se debe trabajar para poder sacar adelante un proyecto de diseño e implementación software teniendo conciencia del mismo desde su gestación en la captura de requisitos hasta su implementación lo que considero que me será de gran utilidad en mi futura carrera profesional puesto que me ha permitido adquirir experiencia en cuanto a la negociación de requisitos con el usuario se refiere y en cuanto a la presentación de la solución adoptada como respuesta a sus necesidades.

Finalizadas las conclusiones extraídas de la realización del proyecto, procedemos a continuación a presentar una serie de líneas futuras que pudieran y debieran ser llevadas a cabo en posteriores versiones de la herramienta. Como ya se ha repetido en varias ocasiones a lo largo de la presente memoria descriptiva, el alcance del proyecto actual consistía en la realización del análisis y diseño de un sistema informático de ayuda para la gestión de información manejada por la Asociación Navarra de Síndrome de Down elaborando así mismo un pequeño prototipo a modo de familiarización con la herramienta que se implementará en un futuro. Por este motivo, la ampliación fundamental y más evidente para este proyecto consiste en la implementación de todas y cada una de las funcionalidades que debe proporcionar la herramienta para satisfacer los requisitos funcionales exigidos por la Asociación, a excepción de las funcionalidades de menú, gestión de perfiles y personal que ya han sido desarrolladas dentro del prototipo elaborado. En relación al prototipo que se acaba de mencionar se centra la segunda propuesta a tener en cuenta como futura mejora al proyecto y es que tanto en la gestión de menús como en la de perfiles y personal han quedado aspectos a falta de implementar para cumplir los requisitos de la aplicación y aspectos a mejorar para proporcionar un mejor servicio al usuario. Entre dichos aspectos a tener en cuenta en futuras versiones de la aplicación se encuentran los citados a continuación:

- Controlar el tipo de acceso que cada usuario puede tener a cada una de las opciones del menú en función del perfil al que pertenezca. Es decir, garantizar que un usuario cuyo perfil para una determinada opción sea “Sin acceso” no pueda acceder a la misma, mientras que si su perfil es “Lectura” pueda acceder a ella, pero no realizar ninguna modificación sobre la información existente y finalmente que si su perfil es “Escritura” pueda realizar todo tipo de modificaciones sobre la información asociada a la opción del menú accedida.
- Tanto en la creación como en la modificación de perfiles y personal, el botón “Volver”, que se muestra en la página de error presentada cuando se produce un error que impide la grabación de la información tratada, devuelve al usuario a la página de listado bien sea de perfiles o de personal. Lo correcto en estos casos, y lo que se propone como futura mejora, sería que dicho botón devolviera al usuario a la página de introducción de la información manteniendo informados aquellos campos en los que el usuario hubiera introducido datos.
- Tanto en la gestión de perfiles como en la de personal, se debiera mantener visible el criterio de búsqueda introducido por el usuario una vez que la aplicación ha devuelto el listado de resultados coincidentes con la consulta realizada. Así mismo, pero en este caso solo relacionado con la funcionalidad de búsqueda de personal, se debería mejorar el hecho de que, después de acceder al detalle de un personal obtenido en una búsqueda, al pulsar el botón “Volver” se devolviera al usuario a la página de listado manteniendo la consulta realizada inicialmente y no mostrando todos los registros de personal existentes.
- La implementación del menú de la aplicación de manera que sólo los usuarios logados en la herramienta puedan visualizar las diferentes opciones proporcionadas por la misma se ha controlado correctamente siempre que el usuario logado abandone la aplicación empleando el botón de desconexión existente para ello. El hecho de que el abandono de la aplicación se realice cerrando el navegador sin haberse desconectado correctamente provoca que se pierda el control realizado sobre el menú de manera que, al acceder de nuevo a la herramienta, éste se muestra visible inicialmente, a pesar de que el usuario no haya introducido su login y clave de acceso. La corrección de este aspecto deberá abordarse en la continuación del proyecto que se realice en un futuro.
- La navegación a través de las diferentes páginas de la aplicación se ha preparado para funcionar correctamente a través de los botones de acción (grabar, buscar, volver, etc.) existentes para ello en cada una de las citadas páginas. Sin embargo, no se ha controlado que dicha navegación funcione también correctamente si el usuario decide hacer uso de los botones de navegación existentes en el navegador. Es decir, en ocasiones se muestra un error de página Web expirada o no se accede a la página deseada si el usuario emplea los botones de navegación hacia delante y hacia atrás del navegador, aspecto que debería solventarse en el proyecto encargado de finalizar la implementación del proyecto aquí analizado, diseñado y prototipado.


Por otra parte, sería interesante que se tuviera en cuenta para futuras versiones de la herramienta una mejora en la interfaz gráfica de la misma puesto que en el proyecto actual no se ha prestado especial atención a dicho aspecto centrándose únicamente en cumplir los criterios básicos de usabilidad. En este sentido sería interesante mejorar fundamentalmente la presentación de los listados y las pantallas de información y error para hacerlas más amigables para el usuario.

Otra de las líneas futuras a considerar en la continuación y mejora del proyecto actual es la realización de pruebas de carga para comprobar el correcto rendimiento de la aplicación implementada así como la obtención por parte del usuario final de la misma de una validación que garantice que la herramienta implementada cumple con las expectativas esperadas de ella. Por último, debería considerarse tanto el hecho de la puesta en funcionamiento de la aplicación dentro de la red interna existente en la Asociación Navarra de Síndrome de Down como la impartición de una pequeña formación al personal de dicha Asociación sobre el manejo de la herramienta implantada.

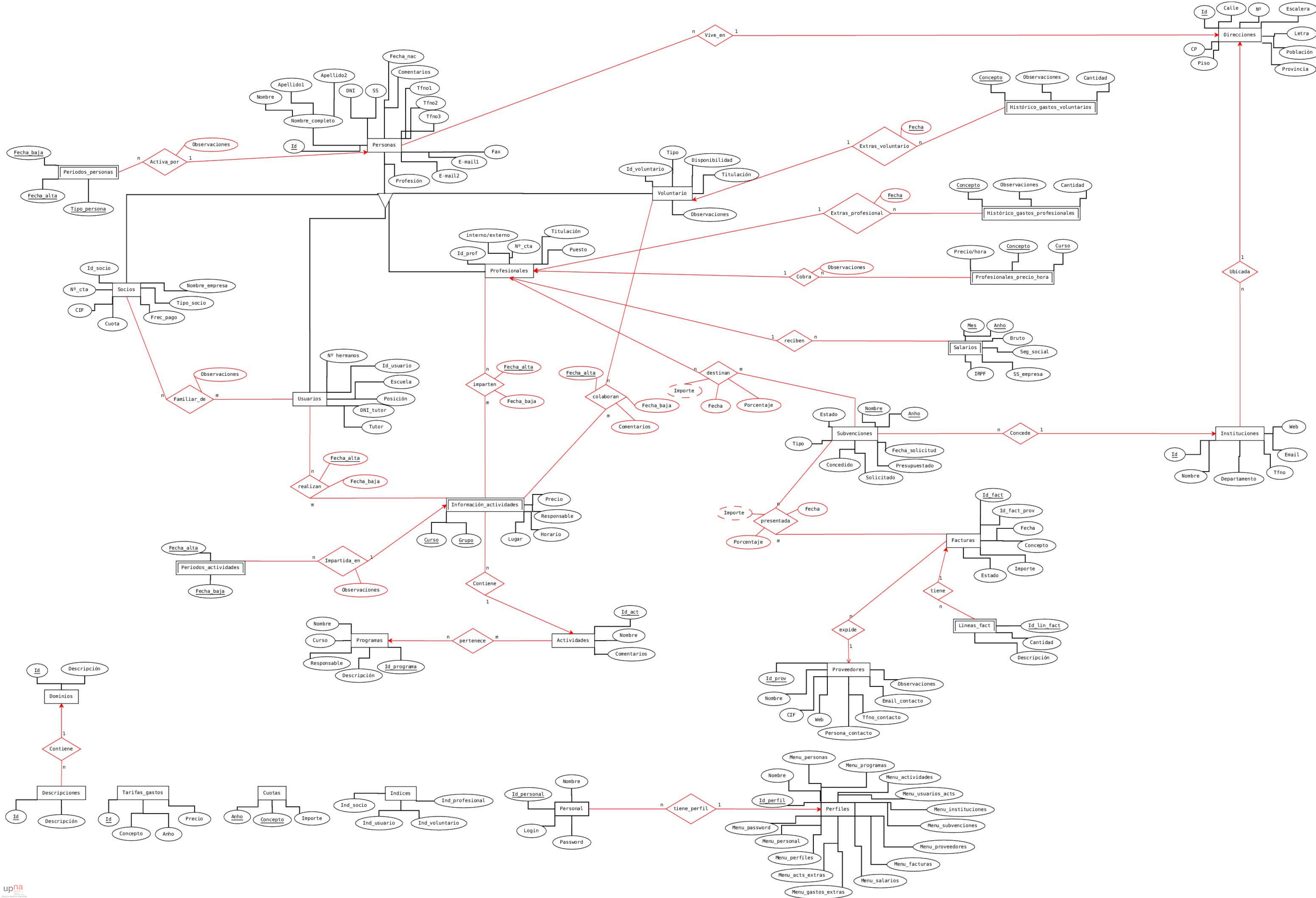
Bibliografía

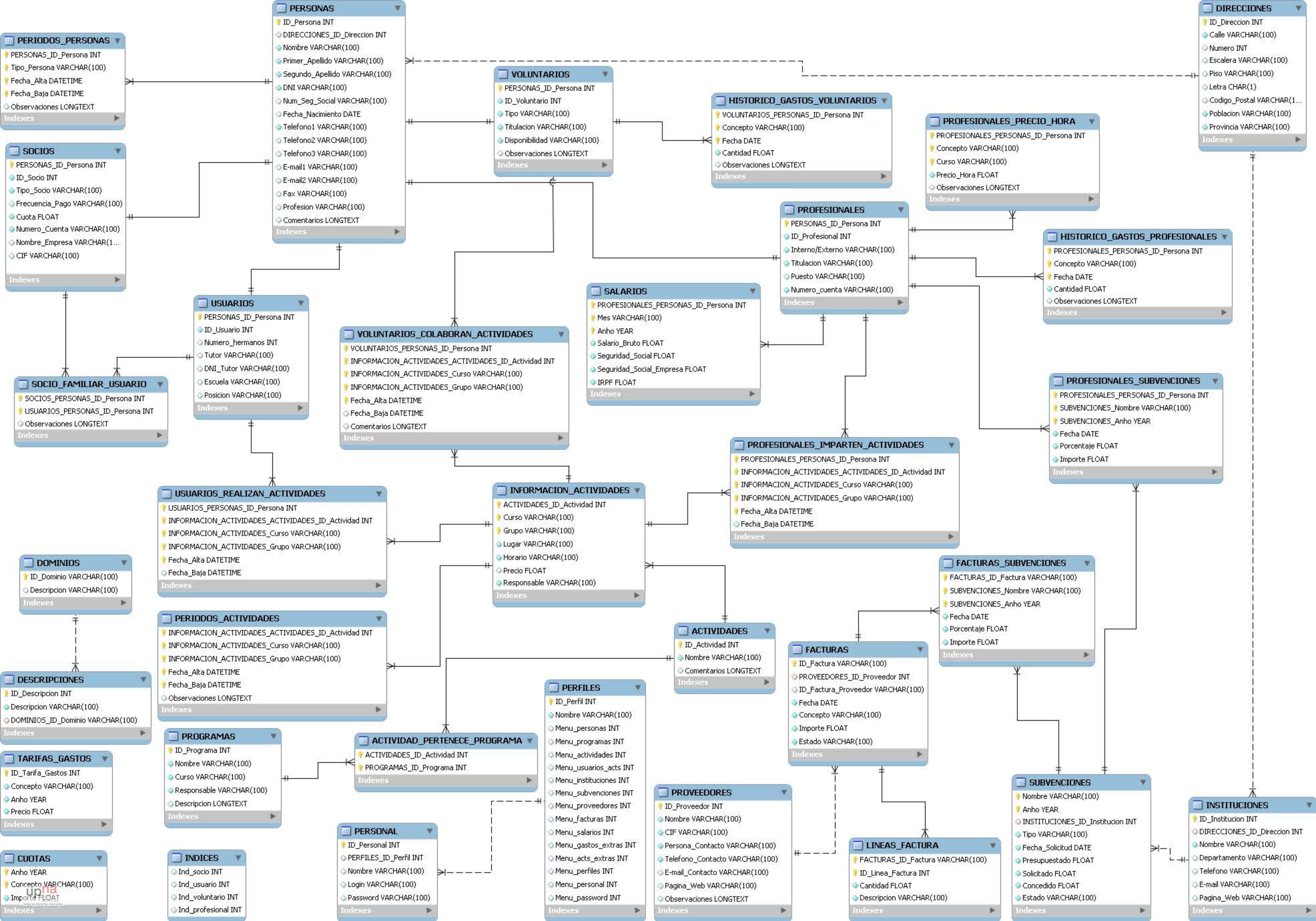
[DownNavarra]	Asociación Navarra de Síndrome de Down, http://www.sindromedownnavarra.org/
[DownEspaña]	Down España, http://www.sindromedown.net/
[MsftAcc]	Microsoft Access, http://office.microsoft.com/es-es/access/
[Def.Req.Hw]	Requisitos Hardware, http://www.alegsa.com.ar/Dic/requerimiento%20de%20hardware.php
[Def.Req.NoFunc]	Requisitos funcionales y no funcionales, http://dis.um.es/~lopezquesada/documentos/Tema%203%20parte%201.ppt#278,22,Requisitos%20funcionales%20y%20no%20funcionales
[Def.REq.NoFuncII]	Requisitos no funcionales http://zoftwar.blogspot.com/2006/11/requisitos-no-funcionales-parte-1-con.html
[Joomla]	CMS Joomla, http://www.joomla.org/
[JoomlaES]	Portal Joomla Spanish, http://www.joomlaspanish.org/
[PHP]	Documentación PHP, http://www.php.net/docs.php
[MySQL]	Servidor MySQL, http://dev.mysql.com/downloads/mysql/
[Apache]	Servidor Apache, http://httpd.apache.org/

[Def.Diag.CasoUso]	Diagrama de Casos de Uso, http://www2.uah.es/jcaceres/uploaded/capsulas/DiagramaCasosDeUso.pdf
[Def.Diag.CasoUsoII]	Diagrama de Casos de Uso, http://www.osmosislatina.com/lenguajes/uml/casos.htm
[Def.Diag.CasoUsoIII]	Diagrama de Casos de Uso, http://www.wikilearning.com/tutorial/desarrollo_orientado_a_objetos_con_uml-diagrama_de_casos_de_uso/6321-5
[Desc.Diag.CasoUso]	Diagrama de Casos de Uso, http://www.slideshare.net/ktyk/uml-casos-de-uso
[Def.DiagramaER]	Modelo de datos, http://basesdedatos.wetpaint.com/page/MODELO+DE+D+ATOS
[Def.DiagramaER.II]	Modelo Entidad - Relación http://es.wikipedia.org/wiki/Modelo_entidad-relaci%C3%B3n
[ElementosER]	Diseño de Bases de Datos relacionales, http://usuarios.multimania.es/cursosgbd/UD4.htm
[Card.Relación]	Modelo Entidad - Relación http://webpcmania.blogcindario.com/2006/06/00272-entidad-relacion-e-r.html
[Def.Arq.Sistema]	Diseño Arquitectura del Sistema, http://academica.cinvestav.mx/sinac/project/node20.html
[Def.Arq.Sw]	Arquitectura de Software, http://es.wikipedia.org/wiki/Arquitectura_software
[Def.Arq.Comp]	Arquitectura de las computadoras, http://www.monografias.com/trabajos17/arquitectura-computadoras/arquitectura-computadoras.shtml
[ApacheDef]	Definición de servidor Apache, http://www.alegsa.com.ar/Dic/apache.php
[ApacheCarac]	Servidor Apache, http://linux.ciberaula.com/articulo/linux_apache_intro/
[InstalarApache]	Instalar Apache en Windows, http://www.masadelante.com/faqs/instalar-apache
[InstalarApacheII]	Instalar Apache en Windows, http://www.desarrolloweb.com/articulos/1379.php

[MySQLAdmin]	MySQL GUI Tools, http://dev.mysql.com/downloads/gui-tools/5.0.html
[EduJoomla]	CMS Joomla, http://www.edujoomla.es/que-es-joomla
[JoomlaInst]	Instalar Joomla en Windows XP, http://www.gtbyte.net/sitio/manuales/joomlawxp.pdf
[MySQLManager]	EMS SQL Manager for MySQL, http://www.sqlmanager.net/products/mysql/manager
[GUIDef]	Definición de GUI, http://www.alegsa.com.ar/Dic/gui.php
[GUIDefWiki]	Interfaz gráfica de usuario, http://es.wikipedia.org/wiki/Interfaz_gr%C3%A1fica_de_usuario
[FichBBDD]	Victoria Mohedano. Apuntes de la asignatura “ <i>Ficheros y Bases de Datos</i> ”. Universidad Pública de Navarra, Curso 2002-2003.
[MySQLInst]	 C:\Documents and Settings\usuario\Escr
[CMS]	Qué es y cómo funciona un CMS, http://business.bligoo.com/content/view/759532/Que-es-y-como-funciona-un-CMS.html
[CMSWiki]	Sistema de gestión de contenidos, http://es.wikipedia.org/wiki/Sistema_de_gesti%C3%B3n_de_contenidos
[WebEmp]	Servicios profesionales Joomla, http://www.webempresa.com/creacion-web-joomla.html
[TutoJoomla]	Tutorial Joomla, http://www.juntadeandalucia.es/averroes/~14700079/upload/cursojoomlacep.pdf
[Ext.Joomla]	Centro de Extensiones Joomla Spanish, http://extensiones.joomlaspanish.org/
[FormFacileForms]	Realizar formularios en Joomla con FacileForms, http://www.joomlaar.com.ar/tutoriales/40
[FacileFormsJoomla]	FacileForms para Joomla, http://www.boxsoftware.net/programas/facileforms-for-joomla-1-4-7.asp

[Def.PHP]	Definición de PHP, http://www.alegsa.com.ar/Dic/php.php
[Int.PHP]	Introducción, definición y evolución de PHP, http://php.ciberaula.com/articulo/introduccion_php/
[Carac.PHP]	Características de PHP, http://www.linuxcentro.net/linux/staticpages/index.php?page=CaracteristicasPHP
[ManualPHP]	Manual de PHP, http://www.php.net/manual/es/index.php
[HistoriaPHP]	Breve historia de PHP, http://www.monografias.com/trabajos66/introduccion-php/introduccion-php2.shtml
[MetaMod]	MetaMod, http://www.metamodpro.com/
[Inst.MetaMod]	Instalación MetaMod, http://www.metamodpro.com/metamod/joomla-1-0/installation-1-0
[MundoJoomla]	Empleo módulo MetaMod, http://mundojoomla.org/tutoriales/boton-dinamico-loginlogout?start=2





SISTEMA INFORMÁTICO DE GESTIÓN WEB PARA LA ASOCIACIÓN NAVARRA DE SÍNDROME DE DOWN

Elena Leoz Laquidain

José Javier Astrain Escola

Pamplona, 25 de Noviembre de 2010

Índice de contenidos

- Introducción
- Análisis
- Diseño
- Implementación
- Conclusiones

Líneas futuras

Introducción

- Asociación Navarra de Síndrome de Down
 - Entidad sin ánimo de lucro
 - Mejorar calidad de vida y defender derechos
- Objetivo
 - Inclusión social e igualdad de las personas con Síndrome de Down
- Actividades
 - Mentalización y sensibilización de la población
 - Específicas para las personas con Síndrome de Down

Introducción

- Estado del arte

- BBDD Access
- Hojas de cálculo
- Documentos de texto

} Información desestructurada e inconsistente

- Objetivo

- Análisis y diseño de una aplicación Web para la Asociación
- Implementación prototipo de familiarización con la aplicación

Introducción

- Planteamiento de la solución
 - Herramienta multiusuario con acceso controlado mediante nombre de usuario y contraseña
 - Funcionamiento sencillo
 - Organizada en los siguientes apartados
 - Personas
 - Programas
 - Actividades
 - Usuarios-Actividades
 - Instituciones
 - Subvenciones
 - Proveedores
 - Facturas
 - Salarios
 - Gastos extraordinarios
 - Actividades extraordinarias
 - Perfiles
 - Personal
 - Cambio password

Introducción

- Metodología de trabajo
 - Análisis
 - Toma de conocimiento
 - Definición de requisitos y alcance del proyecto
 - Diagrama Entidad-Relación
 - Análisis orientado a objetos
 - Diagramas de casos de uso
 - Diseño
 - Diseño e implementación BBDD
 - Diseño de la interfaz
 - Implementación del prototipo

Análisis - Requisitos

- Requisitos de interfaz
 - Interfaz de usuario sencilla, fácil de entender y utilizar
 - Diseño único
 - Menú siempre visible
 - Funcionalidad ventana central
 - Situación botones de acción
- Requisitos funcionales
 - Descripción formal de la funcionalidad de la herramienta

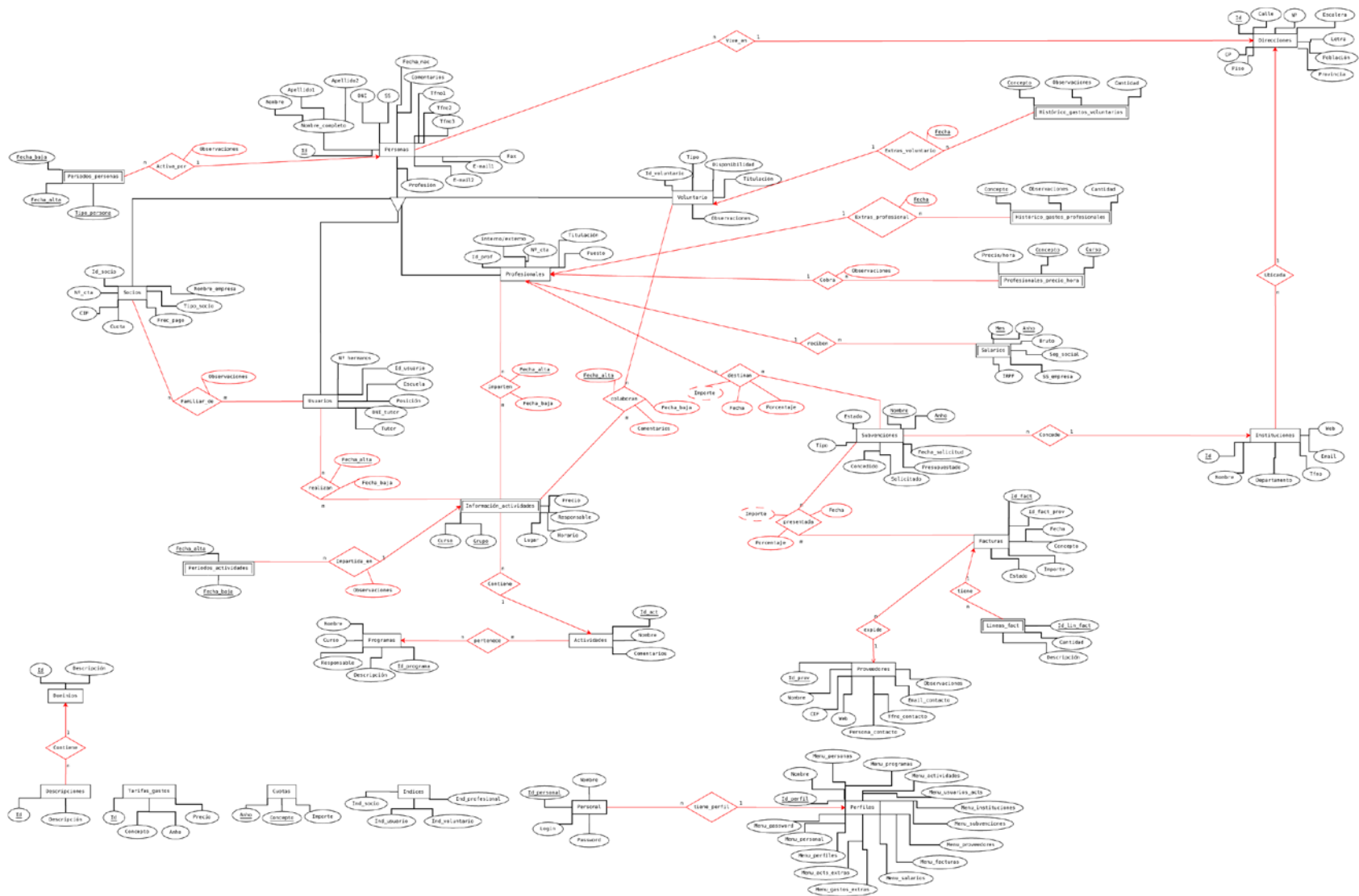
Análisis - Requisitos

- Requisitos no funcionales

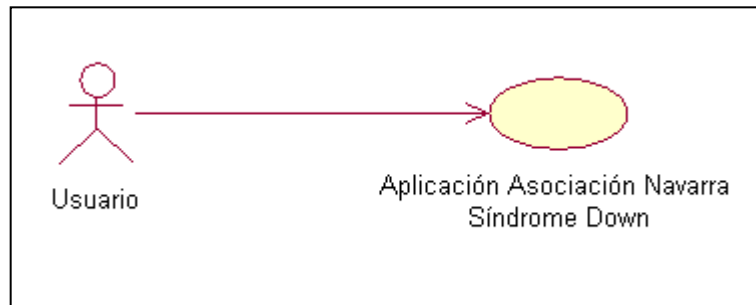
- Cumplimiento LOPD
- Sistema Operativo Windows
- Gestor de Contenidos Joomla
- Lenguaje de programación PHP
- Base de Datos MySQL

} Requisitos Hardware

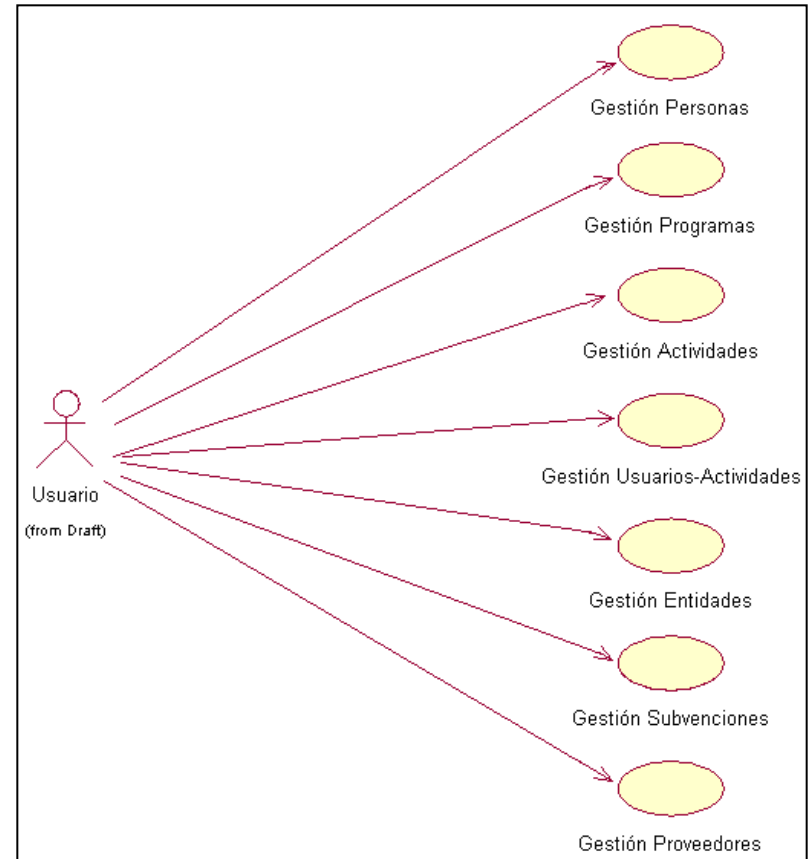
Análisis – Diagrama E/R



Análisis – Diagramas de Casos de Uso

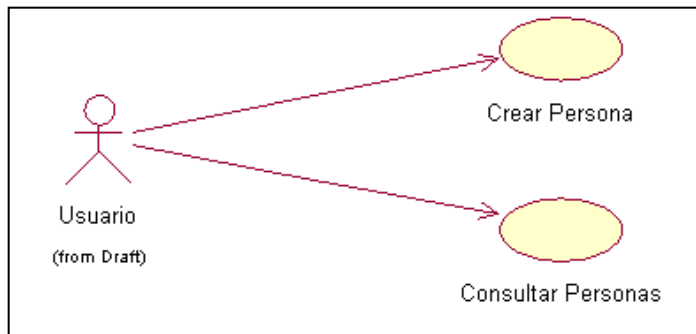


Acceso aplicación

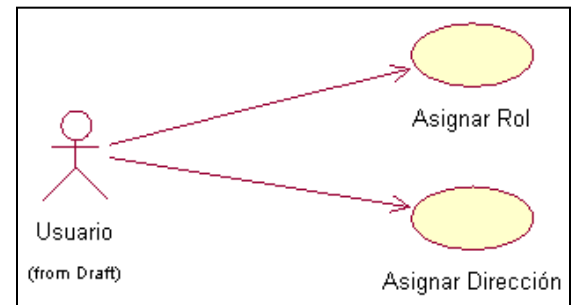


Menú aplicación

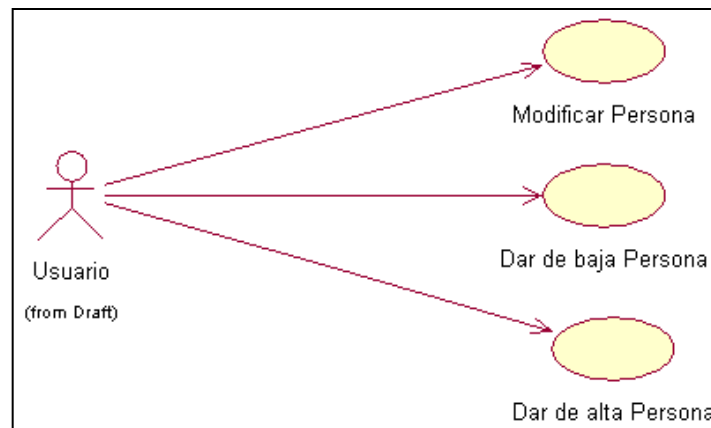
Análisis – Diagramas de Casos de Uso



Gestión Personas

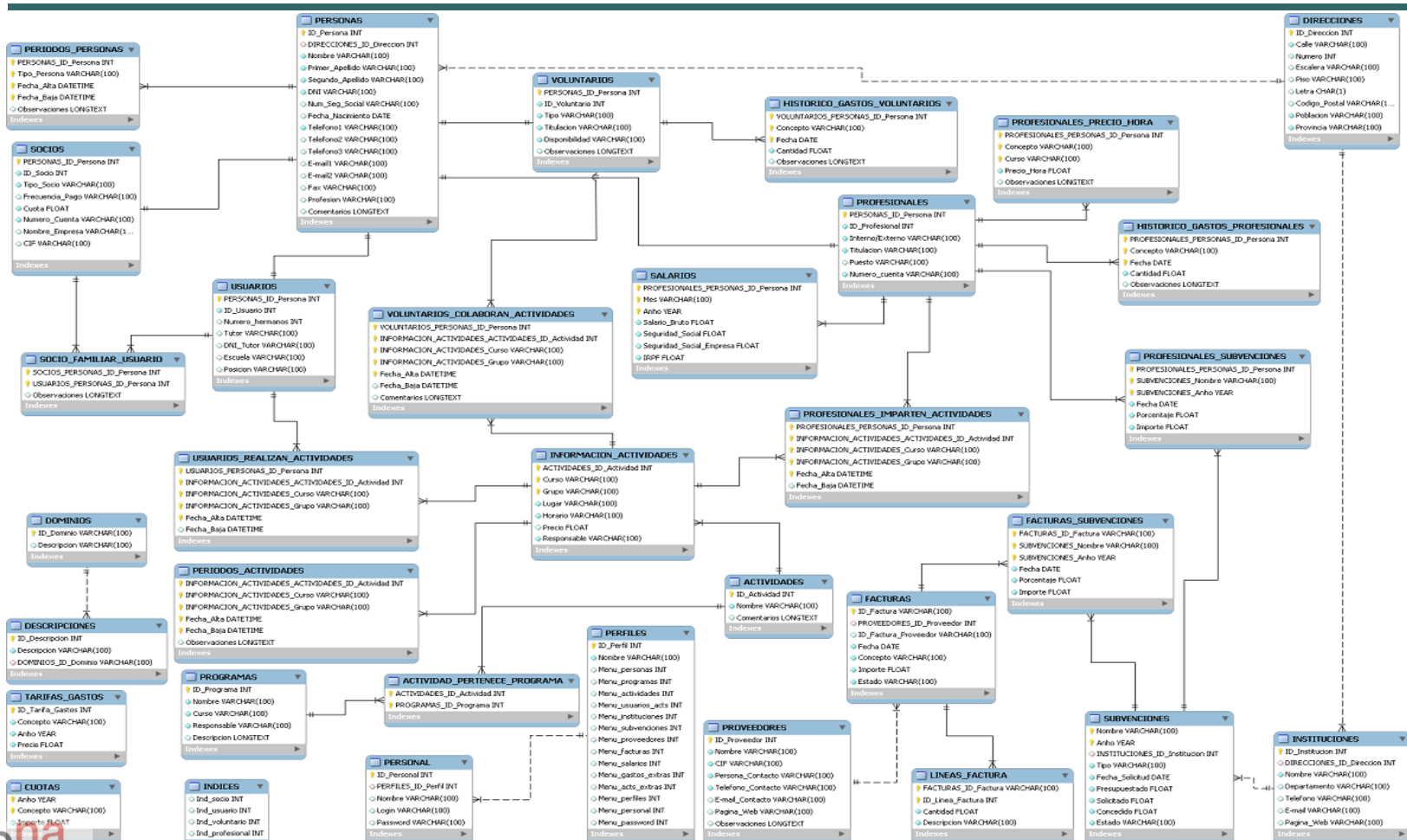


Crear Persona

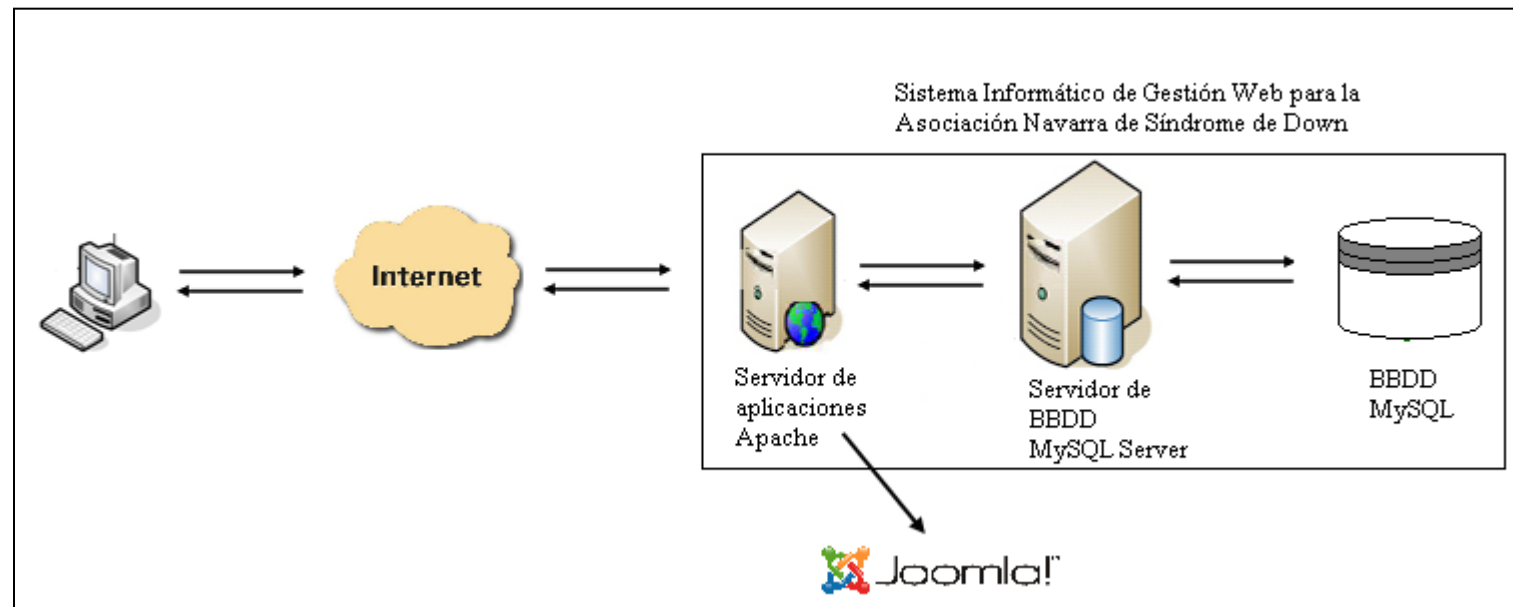


Consultar Personas

Diseño – Base de Datos



Diseño – Arquitectura de la herramienta



Diseño – Interfaz

- Interfaz pantalla listado información

The screenshot displays the 'DOWN Naverre' web application. The header features the logo 'DOWN Naverre' and the text 'Asociación Síndrome de Down'. Below the header, a navigation bar shows 'Asociación Síndrome de Down' and 'Personal'. A yellow sidebar on the left contains a 'Menú' with various options: Personas, Programas, Actividades, Usuarios-Actividades, Entidades, Subvenciones, Proveedores, Facturas, Salarios, Gastos extraordinarios, Actividades extraordinarias, Perfiles, Personal, and Cambio password. The main content area is titled 'Menú >> Personal' and includes a 'Desconectar' button. It contains a form for adding or editing a user with fields for 'Nombre', 'Login', and 'Perfil', along with 'Crear' and 'Buscar' buttons. Below the form is a 'Listado Personal' table with columns for 'Nombre', 'Login', and 'Perfil'. The table lists several users, including 'Superadministrador', 'María Jiménez', 'Alberto Sanz', 'Carmen Marchena', and 'Javier Amatriain'. At the bottom of the page, there is a navigation bar with '<< Comienzo < Anterior Siguiente > Fin >>'.

DOWN Naverre
Asociación Síndrome de Down

Asociación Síndrome de Down Personal

Menú

- Personas
- Programas
- Actividades
- Usuarios-Actividades
- Entidades
- Subvenciones
- Proveedores
- Facturas
- Salarios
- Gastos extraordinarios
- Actividades extraordinarias
- Perfiles
- Personal
- Cambio password

Menú >> Personal

Nombre

Login

Perfil

Crear

Buscar

Desconectar

Listado Personal

Nombre	Login	Perfil
Superadministrador	SUPERADMINISTRADOR	Superadministrador
María Jiménez	MJIMENEZ	Gestor
Alberto Sanz	ASANZ	Monitor
Carmen Marchena	CMARCHENA	Informático
Javier Amatriain	JAMATRIAIN	Contable

<< Comienzo < Anterior Siguiente > Fin >>

Diseño – Interfaz

The screenshot shows the login page of the 'Asociación Síndrome de Down Navarra' website. The header features the organization's logo, which includes the word 'DOWN' in blue and a yellow smiley face, with 'Navarra' and 'Asociación Síndrome de Down' written below it. A navigation bar below the header shows 'Asociación Síndrome de Down' and 'Personal'. On the left, a yellow sidebar menu lists various options: Personas, Programas, Actividades, Usuarios-Actividades, Entidades, Subvenciones, Proveedores, Facturas, Salarios, Gastos extraordinarios, Actividades extraordinarias, Perfiles, Personal, and Cambio password. The main content area, titled 'Menú >> Personal', contains a login form with fields for 'Nombre', 'Login', 'Password', and 'Perfil' (a dropdown menu). A 'Guardar' button is at the bottom right of the form, and a 'Volver' button is at the bottom left. A 'Desconectar' button is located in the top right corner of the main area.

Interfaz pantalla
introducción datos

The screenshot shows the data modification page of the 'Asociación Síndrome de Down Navarra' website. The header and navigation bar are identical to the login page. The main content area, titled '> Personal', contains a form for editing user data. The fields are: 'Nombre' (filled with 'Javier Amatriain'), 'Login' (filled with 'JAMATRIAIN'), 'Password' (filled with dots), and 'Perfil' (a dropdown menu filled with 'Contable'). There are 'Borrar' and 'Guardar' buttons at the bottom right of the form, and a 'Volver' button at the bottom left. A 'Desconectar' button is in the top right corner.

Interfaz pantalla modificación datos

Implementación

- Elementos necesarios
 - Lenguaje de programación PHP
 - Base de Datos MySQL
 - Gestor de Contenidos Joomla
 - Framework creación y administración contenidos
 - Permite crear, clasificar y publicar información en una página Web
 - Independencia contenido e interfaz
 - Elementos básicos: plantillas, componentes, módulos y mambots
 - Componente FacileForms
 - Herramienta para la creación directa y sencilla de formularios
 - Tablas propias en la BBDD del Gestor de Contenidos
 - Permite introducción de código fuente propio en PHP

Implementación

- Procedimiento programación
 - Construcción formularios y páginas
 - Implementación funcionalidad específica FacileForms
 - Script de inicialización
 - Antes del formulario
 - Scripts inicialización elementos página
 - Después del formulario
 - Scripts botones página
 - Antes de enviar
 - Script enviado
 - Después de enviar

Implementación

- Ejemplo procedimiento programación Gestión Personal
 - Inicialización listado personal y búsqueda perfil
 - Código asociado al botón de Crear o Modificar
 - Inicialización elementos nombre, login, password y perfil
 - Código asociado al botón de Guardar, Borrar
 - Código script "Antes de enviar"
 - Código asociado al botón de Buscar
 - Código script "Antes de enviar"

Implementación

- Ejemplo procedimiento programación Menú
 - Creación menú vacío módulo estándar menús Joomla
 - Creación menú opciones aplicación módulo estándar menús Joomla
 - Instalación módulo MetaMod
 - Comprobación usuario logado para mostrar menú vacío o menú opciones aplicación

Conclusiones

- Diseño aplicación de gran utilidad
- Tareas análisis y diseño menos costosas y complicadas
- Implementación prototipo tarea ardua y complicada
 - Joomla no sencillo e intuitivo
 - Funcionamiento componente FacileForms
 - Detalle registro información
 - Funcionalidad búsqueda
- Repaso y puesta en práctica de conocimientos
- Conocimiento realidad laboral

Líneas futuras

- Implementación funcionalidades requisitos funcionales
- Mejorar implementación prototipo
 - Control de accesos opciones menú
 - Mantener información introducida por el usuario
 - Mantener visible criterio de búsqueda
 - Controlar cierre de sesión al cerrar el navegador
 - Controlar navegación a través del navegador

Líneas futuras

- Mejorar interfaz gráfica
- Realización pruebas de carga
- Implantación herramienta
- Formación usuarios

SISTEMA INFORMÁTICO DE GESTIÓN WEB PARA LA ASOCIACIÓN NAVARRA DE SÍNDROME DE DOWN

Elena Leoz Laquidain

José Javier Astrain Escola

Pamplona, 25 de Noviembre de 2010